

## Лабораторна робота №2

### Тема: Породжувальні шаблони

**Мета роботи:** навчитися реалізовувати породжувальні шаблони проєктування

### Хід роботи:

#### Завдання 0: Підготовка до виконання завдання

1. Створити директорію **lab-2** в Вашому репозиторії з лабораторками.
2. Перейти до Завдання 1 😊

#### Завдання 1: Фабричний метод.

1. Напишіть систему класів для реалізації функціоналу створення різних типів підписок для відео провайдера.
2. Кожна з підписок повинна мати щомісячну плату, мінімальний період підписки та список каналів й інших можливостей.
3. Види підписок: **DomesticSubscription, EducationalSubscription, PremiumSubscription.**
4. Придбати (тобто створити) підписку можна за допомогою трьох різних класів: **WebSite, MobileApp, ManagerCall**, кожен з них має реалізувати свою логіку створення підписок.
5. Покажіть правильність роботи свого коду запустивши його в головному методі програми.
6. Підготуйте діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>, експортуйте та завантажте її до репозиторія.

### Лістинг програми:

```
Task1
├── classes
│   └── creators
│       ├── ICreator.php
│       ├── ManagerCall.php
│       ├── MobileApp.php
│       └── WebSite.php
│   └── subscriptions
│       ├── DomesticSubscription.php
│       ├── EducationalSubscription.php
│       ├── ISubscription.php
│       └── PremiumSubscription.php
├── autoload.php
├── confirm_subscription.php
├── index.html
└── process_subscription.php
```

					ДУ «Житомирська політехніка».22.121.16.000 – Лр2		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Кохан Т.О			Звіт з лабораторної роботи №2	Лім.	Арк.
Перевір.		Фант М. О.					Аркушів
Керівник							1
Н. контр.						ФІКТ Гр. ІПЗ-22-З[1]	
Зав. каф.		Морозов А.В.					
							21

## ICreator.php:

```
<?php

namespace classes\creators;
interface ICreator
{
    public function createSubscription($type);
}
```

## ManagerCall.php:

```
<?php

namespace classes\creators;

use classes\subscriptions\DomesticSubscription;
use classes\subscriptions\EducationalSubscription;
use classes\subscriptions\PremiumSubscription;

require_once 'ICreator.php';
class ManagerCall implements ICreator {
    public function createSubscription($type) {
        switch ($type) {
            case 'domestic':
                return new DomesticSubscription();
            case 'educational':
                return new EducationalSubscription();
            case 'premium':
                return new PremiumSubscription();
            default:
                throw new \Exception("Unknown subscription type");
        }
    }
}
```

## MobileApp.php:

```
<?php

namespace classes\creators;

use classes\subscriptions\DomesticSubscription;
use classes\subscriptions\EducationalSubscription;
use classes\subscriptions\PremiumSubscription;

require_once 'ICreator.php';
class MobileApp implements ICreator {
    public function createSubscription($type) {
        switch ($type) {
            case 'domestic':
                return new DomesticSubscription();
            case 'educational':
                return new EducationalSubscription();
            case 'premium':
                return new PremiumSubscription();
            default:
                throw new \Exception("Unknown subscription type");
        }
    }
}
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

## WebSite.php:

```
<?php

namespace classes\creators;

use classes\subscriptions\DomesticSubscription;
use classes\subscriptions\EducationalSubscription;
use classes\subscriptions\PremiumSubscription;

require_once 'ICreator.php';
class WebSite implements ICreator {
    public function createSubscription($type) {
        switch ($type) {
            case 'domestic':
                return new DomesticSubscription();
            case 'educational':
                return new EducationalSubscription();
            case 'premium':
                return new PremiumSubscription();
            default:
                throw new \Exception("Unknown subscription type");
        }
    }
}
```

## DomesticSubscription.php:

```
<?php

namespace classes\subscriptions;

require_once 'ISubscription.php';
class DomesticSubscription implements ISubscription
{
    public function getMonthlyFee()
    {
        return 10;
    }

    public function getMinimumPeriod()
    {
        return 6;
    }

    public function getChannels()
    {
        return ['Новини', 'Спорт', 'Розваги'];
    }
}
```

## EducationalSubscription.php:

```
<?php

namespace classes\subscriptions;

require_once 'ISubscription.php';
class EducationalSubscription implements ISubscription {
    public function getMonthlyFee() {
        return 5;
    }
}
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```
public function getMinimumPeriod() {
    return 12;
}

public function getChannels() {
    return ['Наукові', 'Історичні', 'Документальні'];
}
}
```

### PremiumSubscription.php:

```
<?php

namespace classes\subscriptions;

require_once 'ISubscription.php';
class PremiumSubscription implements ISubscription {
    public function getMonthlyFee() {
        return 20;
    }

    public function getMinimumPeriod() {
        return 1;
    }

    public function getChannels() {
        return ['Фільми', 'Premium Sports', 'Ексклюзивні серіали'];
    }
}
```

### ISubscription.php:

```
<?php

namespace classes\subscriptions;
interface ISubscription
{
    public function getMonthlyFee();

    public function getMinimumPeriod();

    public function getChannels();
}
```

### autoload.php:

```
<?php
/**
 * Функція для автопідключення класів.
 *
 * @param $class string Назва класу (передається автоматично)
 * @return void
 */
function autoload($class){
    $path = str_replace('\\', '/', $class) . ".php";
    if(is_file($path))
        require_once ($path);
}
spl_autoload_register('autoload');
```

### confirm\_subscription.php:

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

<?php
require_once 'autoload.php';

session_start();

if (!isset($_SESSION['subscription']) || !isset($_SESSION['subscriptionType'])
|| !isset($_SESSION['purchaseMethod'])) {
    header('Location: index.html');
    exit;
}

$subscription = unserialize($_SESSION['subscription']);
$subscriptionType = $_SESSION['subscriptionType'];
$purchaseMethod = $_SESSION['purchaseMethod'];

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST['confirm'])) {
        $message = "Ви успішно створили підписку на " .
ucfirst($subscriptionType) . " . Метод створення: " . ucfirst($purchaseMethod) .
".";
        echo "<script type='text/javascript'>alert('$message');";
window.location.href = 'index.html';</script>";
        session_destroy();
    } elseif (isset($_POST['back'])) {
        header('Location: index.html');
        exit;
    }
} else {
    echo "<div style='max-width: 600px; margin: 0 auto; padding: 20px; border:
1px solid darkgray; border-radius: 5px; background-color: beige;'>";
    echo "<h2 style='text-align: center;'>Опис даної підписки:</h2>";
    echo "<p><strong>Щомісячна оплата:</strong> $" . $subscription-
>getMonthlyFee() . "</p>";
    echo "<p><strong>Мінімальний період:</strong> " . $subscription-
>getMinimumPeriod() . " місяців</p>";
    echo "<p><strong>Доступні канали:</strong> " . implode(', ', $subscription-
>getChannels()) . "</p>";

    echo '<div style="text-align: center; margin-top: 20px;">';
    echo '<form method="post" action="confirm_subscription.php">';
    echo '<input type="submit" name="confirm" value="Оформити підписку"
style="background-color: green; color: white; border: none; border-radius: 3px;
padding: 10px; margin-right: 10px;">';
    echo '<input type="submit" name="back" value="Назад" style="background-
color: grey; color: black; border: none; border-radius: 3px; padding: 10px;">';
    echo '</form>';
    echo '</div>';
    echo "</div>";
}

```

### process\_subscription.php:

```

<?php

require_once 'autoload.php';

use classes\creators\WebSite;
use classes\creators\MobileApp;
use classes\creators\ManagerCall;

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $subscriptionType = $_POST['subscriptionType'];
    $purchaseMethod = $_POST['purchaseMethod'];

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$creator = null;
switch ($purchaseMethod) {
    case 'website':
        $creator = new WebSite();
        break;
    case 'mobileApp':
        $creator = new MobileApp();
        break;
    case 'managerCall':
        $creator = new ManagerCall();
        break;
    default:
        throw new Exception("Невідомий метод створення");
}

$subscription = $creator->createSubscription($subscriptionType);

session_start();
$_SESSION['subscription'] = serialize($subscription);
$_SESSION['subscriptionType'] = $subscriptionType;
$_SESSION['purchaseMethod'] = $purchaseMethod;

header('Location: confirm_subscription.php');
exit;
}

```

### index.html:

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width, user-scalable=no, initial-scale=1.0,
maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Main Page</title>
    <style>
        form {
            max-width: 400px;
            margin: 0 auto;
            padding: 20px;
            border: 1px solid bisque;
            border-radius: 5px;
            background-color: beige;
        }
        h1 {
            text-align: center;
        }
        label {
            display: block;
            margin-bottom: 10px;
        }
        select, input[type="submit"] {
            width: 100%;
            padding: 10px;
            margin-bottom: 15px;
            border: 1px solid ghostwhite;
            border-radius: 3px;
            box-sizing: border-box;
        }
        input[type="submit"] {
            background-color: gray;
            color: white;
            border: none;
            cursor: pointer;
        }
    </style>

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

```

</style>
</head>
<body>
<h1>Оберіть вашу підписку</h1>
<form action="process_subscription.php" method="post">
  <label for="subscriptionType">Тип підписки:</label>
  <select name="subscriptionType" id="subscriptionType">
    <option value="domestic">Domestic</option>
    <option value="educational">Educational</option>
    <option value="premium">Premium</option>
  </select><br><br>

  <label for="purchaseMethod">Метод створення:</label>
  <select name="purchaseMethod" id="purchaseMethod">
    <option value="website">WebSite</option>
    <option value="mobileApp">MobileApp</option>
    <option value="managerCall">ManagerCall</option>
  </select><br><br>

  <input type="submit" value="Далі">
</form>
</body>
</html>

```

## Результат виконання програми:

### Оберіть вашу підписку

Тип підписки:

Domestic

Метод створення:

WebSite

Далі

#### Опис даної підписки:

Щомісячна оплата: \$10

Мінімальний період: 6 місяців

Доступні канали: Новини, Спорт, Розваги

Оформити підписку

Назад

#### Повідомлення з localhost

Ви успішно створили підписку на Domestic .Метод створення: Website.

ОК

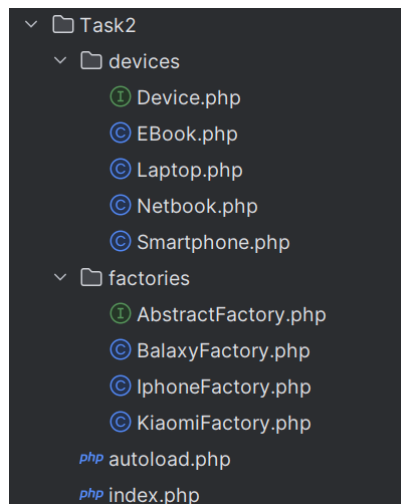
		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

## Завдання 2: Абстрактна фабрика.

1. Створіть фабрику виробництва техніки.
2. На фабриці мають створюватися різні девайси (наприклад, Laptop, Netbook, EBook, Smartphone) для різних брендів (IPhone, Kiaomi, Balaxy).
3. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

4. Підготуйте діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>, експортуйте та завантажте її до репозиторія.

### Лістинг програми:



### Device.php:

```
<?php
namespace devices;

interface Device {
    public function displayInfo();
}
```

### Ebook.php:

```
<?php
namespace devices;

require_once 'Device.php';
class EBook implements Device {
    private $brand;

    public function __construct($brand) {
        $this->brand = $brand;
    }
}
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



```

        public function displayInfo() {
            echo $this->brand . " EBook\n";
        }
    }
}

```

## Laptop.php:

```

<?php

namespace devices;

require_once 'Device.php';
class Laptop implements Device {
    private $brand;

    public function __construct($brand) {
        $this->brand = $brand;
    }

    public function displayInfo() {
        echo $this->brand . " Laptop\n";
    }
}

```

## Netbook.php:

```

<?php

namespace devices;

require_once 'Device.php';
class Netbook implements Device {
    private $brand;

    public function __construct($brand) {
        $this->brand = $brand;
    }

    public function displayInfo() {
        echo $this->brand . " Netbook\n";
    }
}

```

## Smartphone.php:

```

<?php

namespace devices;

require_once 'Device.php';
class Smartphone implements Device {
    private $brand;

    public function __construct($brand) {
        $this->brand = $brand;
    }

    public function displayInfo() {
        echo $this->brand . " Smartphone\n";
    }
}

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

## AbsctractFactory.php:

```
<?php

namespace factories;

use Devices\Device;

interface AbstractFactory {
    public function createLaptop(): Device;
    public function createNetbook(): Device;
    public function createEBook(): Device;
    public function createSmartphone(): Device;
}
```

## BalaxyFactory.php:

```
<?php

namespace factories;

use devices\Laptop;
use devices\Netbook;
use devices\EBook;
use devices\Smartphone;
use devices\Device;
class BalaxyFactory implements AbstractFactory {
    public function createLaptop(): Device {
        return new Laptop('Balaxy');
    }

    public function createNetbook(): Device {
        return new Netbook('Balaxy');
    }

    public function createEBook(): Device {
        return new EBook('Balaxy');
    }

    public function createSmartphone(): Device {
        return new Smartphone('Balaxy');
    }
}
```

## IphoneFactory.php:

```
<?php

namespace factories;

use devices\Laptop;
use devices\Netbook;
use devices\EBook;
use devices\Smartphone;
use devices\Device;
class IPhoneFactory implements AbstractFactory {
    public function createLaptop(): Device {
        return new Laptop('IPhone');
    }

    public function createNetbook(): Device {
        return new Netbook('IPhone');
    }
}
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

```

        public function createEBook(): Device {
            return new EBook('IPhone');
        }

        public function createSmartphone(): Device {
            return new Smartphone('IPhone');
        }
    }
}

```

## KiaomiFactory.php:

```

<?php

namespace factories;

use devices\Laptop;
use devices\Netbook;
use devices\EBook;
use devices\Smartphone;
use devices\Device;

class KiaomiFactory implements AbstractFactory {
    public function createLaptop(): Device {
        return new Laptop('Kiaomi');
    }

    public function createNetbook(): Device {
        return new Netbook('Kiaomi');
    }

    public function createEBook(): Device {
        return new EBook('Kiaomi');
    }

    public function createSmartphone(): Device {
        return new Smartphone('Kiaomi');
    }
}

```

## index.php:

```

<?php

require_once 'autoload.php';

use factories\IPhoneFactory;
use factories\KiaomiFactory;
use factories\BalaxyFactory;

function main() {
    $iphoneFactory = new IPhoneFactory();
    $kiaomiFactory = new KiaomiFactory();
    $balaxyFactory = new BalaxyFactory();

    $iphoneLaptop = $iphoneFactory->createLaptop();
    $iphoneNetbook = $iphoneFactory->createNetbook();
    $iphoneEBook = $iphoneFactory->createEBook();
    $iphoneSmartphone = $iphoneFactory->createSmartphone();

    $kiaomiLaptop = $kiaomiFactory->createLaptop();
    $kiaomiNetbook = $kiaomiFactory->createNetbook();
    $kiaomiEBook = $kiaomiFactory->createEBook();
    $kiaomiSmartphone = $kiaomiFactory->createSmartphone();
}

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

```

$balaxyLaptop = $balaxyFactory->createLaptop();
$balaxyNetbook = $balaxyFactory->createNetbook();
$balaxyEBook = $balaxyFactory->createEBook();
$balaxySmartphone = $balaxyFactory->createSmartphone();

echo "<h2>IPhone Devices:</h2>";
echo "<ul>";
echo "<li>" . $iphoneLaptop->displayInfo() . "</li>";
echo "<li>" . $iphoneNetbook->displayInfo() . "</li>";
echo "<li>" . $iphoneEBook->displayInfo() . "</li>";
echo "<li>" . $iphoneSmartphone->displayInfo() . "</li>";
echo "</ul>";

echo "<h2>Kiaomi Devices:</h2>";
echo "<ul>";
echo "<li>" . $kiaomiLaptop->displayInfo() . "</li>";
echo "<li>" . $kiaomiNetbook->displayInfo() . "</li>";
echo "<li>" . $kiaomiEBook->displayInfo() . "</li>";
echo "<li>" . $kiaomiSmartphone->displayInfo() . "</li>";
echo "</ul>";

echo "<h2>Balaxy Devices:</h2>";
echo "<ul>";
echo "<li>" . $balaxyLaptop->displayInfo() . "</li>";
echo "<li>" . $balaxyNetbook->displayInfo() . "</li>";
echo "<li>" . $balaxyEBook->displayInfo() . "</li>";
echo "<li>" . $balaxySmartphone->displayInfo() . "</li>";
echo "</ul>";
}

main();

```

## Результат виконання програми:

### IPhone Devices:

- Iphone Laptop
- Iphone Netbook
- Iphone EBook
- Iphone Smartphone

### Kiaomi Devices:

- Kiaomi Laptop
- Kiaomi Netbook
- Kiaomi EBook
- Kiaomi Smartphone

### Balaxy Devices:

- Balaxy Laptop
- Balaxy Netbook
- Balaxy EBook
- Balaxy Smartphone

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

### Завдання 3: Одинак.

1. Створіть клас ***Authenticator*** таким чином, щоб бути впевненим, що цей клас може створити лише один екземпляр, незалежно від кількості потоків і класів, що його наслідують.
2. Покажіть правильність роботи свого коду запусивши його в головному методі програми.

#### Лістинг програми:

##### index.php:

```
<?php
class Authenticator {
    private static $instance = null;
    private function __construct() {}

    public static function getInstance() {
        if (self::$instance === null) {
            self::$instance = new self();
        }
        return self::$instance;
    }

    public function authenticate($username, $password) {
        return ($username === 'admin' && $password === 'admin');
    }
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    $authenticator = Authenticator::getInstance();
    $isAuthenticated = $authenticator->authenticate($username, $password);

    if ($isAuthenticated) {
        echo "Успішно автентифіковано!";
    } else {
        echo "Невірний логін або пароль.";
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Main</title>
</head>
<body>
<h2>Форма автентифікації</h2>
<form method="post" action="">
    <label for="username">Логін:</label>
    <input type="text" id="username" name="username" required><br><br>
    <label for="password">Пароль:</label>
    <input type="password" id="password" name="password" required><br><br>
    <input type="submit" value="Увійти">
</form>
</body>
</html>
```

## Результат виконання програми:

Успішно автентифіковано!

## Форма автентифікації

Логін:

Пароль:

### Завдання 4: Прототип.

1. Створіть клас **Virus**. Він повинен містити вагу, вік, ім'я, вид і масив дітей, екземплярів **Virus**.
2. Створіть екземпляри для цілого "сімейства" вірусів (мінімум три покоління).
3. За допомогою шаблону Прототип реалізуйте можливість клонування наявних вірусів.
4. При клонуванні віруса-батька повинні клонуватися всі його діти.
5. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

## Лістинг програми:

### Virus.php:

```
<?php
class Virus
{
    private $name;
    private $species;
    private $children = [];

    public function __construct($name, $species) {
        $this->name = $name;
        $this->species = $species;
    }

    public function addChild(Virus $child) {
        $this->children[] = $child;
    }

    public function clone() {
        return clone $this;
    }

    public function getName() {
        return $this->name;
    }
}
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    public function getSpecies() {
        return $this->species;
    }

    public function getChildren() {
        return $this->children;
    }
}

```

## index.php:

```

<?php
require_once 'Virus.php';

$virus1 = new Virus("Virus1", "Coronavirus");
$virus2 = new Virus("Virus2", "Influenza");
$virus3 = new Virus("Virus3", "Ebola");

$virus1->addChild(new Virus("Child1", "Coronavirus"));
$virus1->addChild(new Virus("Child2", "Coronavirus"));

$virus2->addChild(new Virus("Child3", "Influenza"));
$virus2->addChild(new Virus("Child4", "Influenza"));

$virus3->addChild(new Virus("Child5", "Ebola"));
$virus3->addChild(new Virus("Child6", "Ebola"));

$clone1 = $virus1->clone();
$clone2 = $virus2->clone();
$clone3 = $virus3->clone();

echo "<strong>Клон 1:</strong> " . $clone1->getName() . " (" . $clone1->
    >getSpecies() . ")<br>";
echo "<strong>Діти клону 1:</strong><br>";
foreach ($clone1->getChildren() as $child) {
    echo "- " . $child->getName() . " (" . $child->getSpecies() . ")<br>";
}
echo "<br>";

echo "<strong>Клон 2:</strong> " . $clone2->getName() . " (" . $clone2->
    >getSpecies() . ")<br>";
echo "<strong>Діти клону 2:</strong><br>";
foreach ($clone2->getChildren() as $child) {
    echo "- " . $child->getName() . " (" . $child->getSpecies() . ")<br>";
}
echo "<br>";

echo "<strong>Клон 3:</strong> " . $clone3->getName() . " (" . $clone3->
    >getSpecies() . ")<br>";
echo "<strong>Діти клону 3:</strong><br>";
foreach ($clone3->getChildren() as $child) {
    echo "- " . $child->getName() . " (" . $child->getSpecies() . ")<br>";
}

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

## Результат виконання програми:

**Клон 1:** Virus1 (Coronavirus)

**Діти клону 1:**

- Child1 (Coronavirus)
- Child2 (Coronavirus)

**Клон 2:** Virus2 (Influenza)

**Діти клону 2:**

- Child3 (Influenza)
- Child4 (Influenza)

**Клон 3:** Virus3 (Ebola)

**Діти клону 3:**

- Child5 (Ebola)
- Child6 (Ebola)

### Завдання 5: Будівельник.

1. Створіть клас **HeroBuilder**, який буде створювати персонажа гри, поступово додаючи до нього різні ознаки, наприклад зріст, статуру, колір волосся, очей, одяг, інвентар тощо (можете включити фантазію).
2. Створіть клас **EnemyBuilder**, який буде реалізовувати єдиний інтерфейс з **HeroBuilder**. Відмінністю між ними можуть бути спеціальні методи для творення добра або зла, а також списки добрих і злих справ відповідно.
3. За допомогою свого білдера і класу-директора створіть героя (або героїню) своєї мрії 😊, а також свого найзапеклішого ворога.
4. Зверніть увагу, що Ваші білдери повинні реалізовувати текучий інтерфейс (fluent interface).

5. Покажіть правильність роботи свого коду запустивши його в головному методі програми.
6. Підготуйте діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>, експортуйте та завантажте її до репозиторія.

## Лістинг програми:

### Character.php:

```
<?php
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				16
Змн.	Арк.	№ докум.	Підпис	Дата		



```

class Character
{
    private $height;
    private $build;
    private $hairColor;
    private $eyeColor;
    private $clothing;
    private $inventory;
    private $actions = [];

    public function setHeight($height) {
        $this->height = $height;
    }

    public function setBuild($build) {
        $this->build = $build;
    }

    public function setHairColor($hairColor) {
        $this->hairColor = $hairColor;
    }

    public function setEyeColor($eyeColor) {
        $this->eyeColor = $eyeColor;
    }

    public function setClothing($clothing) {
        $this->clothing = $clothing;
    }

    public function setInventory($inventory) {
        $this->inventory = $inventory;
    }

    public function setActions($actions) {
        if (is_array($actions)) {
            $this->actions = $actions;
        } else {
            $this->actions = [$actions];
        }
    }

    public function describe() {
        echo "Зріст: {$this->height}<br>";
        echo "Статура: {$this->build}<br>";
        echo "Колір волосся: {$this->hairColor}<br>";
        echo "Колір очей: {$this->eyeColor}<br>";
        echo "Одяг, який любить носити: {$this->clothing}<br>";
        echo "Знаряддя, яке використовує: {$this->inventory}<br>";
        echo "Сила: " . implode(", ", $this->actions) . "<br>";
    }
}

```

## CharacterBuilder.php:

```

<?php
interface CharacterBuilder {
    public function setActions($actions);
    public function setHeight($height);
    public function setBuild($build);
    public function setHairColor($hairColor);
    public function setEyeColor($eyeColor);
    public function setClothing($clothing);
    public function setInventory($inventory);
}

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

```

        public function buildCharacter();
    }

```

### create\_character.php:

```

<?php
require_once 'HeroBuilder.php';
require_once 'EnemyBuilder.php';

$type = $_POST['type'];
$actions = $_POST['actions'];
$height = $_POST['height'];
$build = $_POST['build'];
$hairColor = $_POST['hair_color'];
$eyeColor = $_POST['eye_color'];
$clothing = $_POST['clothing'];
$inventory = $_POST['inventory'];

if ($type === 'hero') {
    $builder = new HeroBuilder();
} elseif ($type === 'enemy') {
    $builder = new EnemyBuilder();
}

$character = $builder->setActions($actions)
    ->setHeight($height)
    ->setBuild($build)
    ->setHairColor($hairColor)
    ->setEyeColor($eyeColor)
    ->setClothing($clothing)
    ->setInventory($inventory)
    ->buildCharacter();

echo "<h2>Ваш персонаж</h2>";
$character->descri

```

### EnemyBuilder.php:

```

<?php
require_once 'Character.php';
require_once 'CharacterBuilder.php';

class EnemyBuilder implements CharacterBuilder {
    private $character;
    private $actions = [];

    public function __construct() {
        $this->character = new Character();
    }

    public function setActions($actions) {
        $this->actions = (array) $actions;
        return $this;
    }

    public function setHeight($height) {
        $this->character->setHeight($height);
        return $this;
    }

    public function setBuild($build) {

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

```
        $this->character->setBuild($build);
        return $this;
    }

    public function setHairColor($hairColor) {
        $this->character->setHairColor($hairColor);
        return $this;
    }

    public function setEyeColor($eyeColor) {
        $this->character->setEyeColor($eyeColor);
        return $this;
    }

    public function setClothing($clothing) {
        $this->character->setClothing($clothing);
        return $this;
    }

    public function setInventory($inventory) {
        $this->character->setInventory($inventory);
        return $this;
    }

    public function buildCharacter() {
        $this->character->setActions($this->actions);
        return $this->character;
    }
}
```

**HeroBuilder.php:**

```
<?php

require_once 'Character.php';
require_once 'CharacterBuilder.php';

class HeroBuilder implements CharacterBuilder {
    private $character;
    private $actions = [];

    public function __construct() {
        $this->character = new Character();
    }

    public function setActions($actions) {
        $this->actions = (array) $actions;
        return $this;
    }

    public function setHeight($height) {
        $this->character->setHeight($height);
        return $this;
    }

    public function setBuild($build) {
        $this->character->setBuild($build);
        return $this;
    }

    public function setHairColor($hairColor) {
        $this->character->setHairColor($hairColor);
        return $this;
    }
}
```

```
public function setEyeColor($eyeColor) {
    $this->character->setEyeColor($eyeColor);
    return $this;
}

public function setClothing($clothing) {
    $this->character->setClothing($clothing);
    return $this;
}

public function setInventory($inventory) {
    $this->character->setInventory($inventory);
    return $this;
}

public function buildCharacter() {
    $this->character->setActions($this->actions);
    return $this->character;
}
}
```

**index.php:**

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width, initial-scale=1.0">
    <title>Main</title>
    <script>
        function showActions() {
            var characterType = document.getElementById("type").value;
            var actionsSelect = document.getElementById("actions");

            actionsSelect.innerHTML = '';

            if (characterType === "hero") {
                var goodActions = ["Захищати світ від поганців", "Допомагати
іншим", "Турбуватися про природу"];
                for (var i = 0; i < goodActions.length; i++) {
                    var option = document.createElement("option");
                    option.text = goodActions[i];
                    actionsSelect.add(option);
                }
            } else if (characterType === "enemy") {
                var evilActions = ["Знищувати будинки", "Створювати конфлікти",
"Шкодити довкіллю"];
                for (var i = 0; i < evilActions.length; i++) {
                    var option = document.createElement("option");
                    option.text = evilActions[i];
                    actionsSelect.add(option);
                }
            }
        }
    </script>
</head>
<body>
<h2>Створіть свого персонажа</h2>
<form action="create_character.php" method="post">
    <label for="type">Яким персонажем ви хочете бути?</label>
    <select name="type" id="type" onchange="showActions()">
        <option value="hero">Добрим</option>
        <option value="enemy">Злим</option>
    </select>
</form>
```

```

<br><br>
<label for="actions">Оберіть силу персонажа:</label>
<select name="actions" id="actions">
</select>
<br><br>
<label for="height">Зріст:</label>
<input type="text" id="height" name="height">
<br><br>
<label for="build">Статура:</label>
<input type="text" id="build" name="build">
<br><br>
<label for="hair_color">Колір волосся:</label>
<input type="text" id="hair_color" name="hair_color">
<br><br>
<label for="eye_color">Колір очей:</label>
<input type="text" id="eye_color" name="eye_color">
<br><br>
<label for="clothing">Одяг, який любить носити:</label>
<input type="text" id="clothing" name="clothing">
<br><br>
<label for="inventory">Знаряддя, яке використовує:</label>
<input type="text" id="inventory" name="inventory">
<br><br>
<input type="submit" value="Створити">
</form>
</body>
</html>

```

## Результат виконання програми:

### Створіть свого персонажа

Яким персонажем ви хочете бути?

Оберіть силу персонажа:

Зріст:

Статура:

Колір волосся:

Колір очей:

Одяг, який любить носити:

Знаряддя, яке використовує:

### Ваш персонаж

Зріст: 158

Статура: Сильний

Колір волосся: Каштановий

Колір очей: Блакитний

Одяг, який любить носити: Піжами

Знаряддя, яке використовує: Чарівна паличка

Сила: Турбуватися про природу

**Висновок:** під час виконання лабораторної роботи я навчилася реалізовувати породжувальні шаблони проектування.

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		21