

Задача №1

найти и убить файлы от удаленных товаров (или от старого сайта), которые лежат вместе с картинками товаров в одной директории на диске

Решение:

Построим на использовании агента, который будет каждые сутки чистить папку “/upload/iblock/”. Это позволяет нам настроить один раз и забыть об этой проблеме.

1) Сделаем файл /bitrix/php_interface/include/agents.php

```
<?php
function CleanupImages() {
    define("NO_KEEP_STATISTIC", true);
    define("NOT_CHECK_PERMISSIONS", true);
    // Директория, в которой лежат картинки товаров
    $dirPath = $_SERVER['DOCUMENT_ROOT'] . "/upload/iblock";
    // Получить файлы из таблицы "b_file", которые связаны с элементами инфоблоков
    $dbFile = \Bitrix\Main\FileTable::getList([
        'filter' => [
            '=MODULE_ID' => "iblock"
        ],
        'select' => [
            'SUBDIR',
            'FILE_NAME'
        ]
    ]);
    $arFiles = [];
    while ($arFile = $dbFile->fetch()) {
        $arFiles[] = $arFile["FILE_NAME"];
    }
    $rootDir = opendir($dirPath);
    while (($subDirName = readdir($rootDir)) !== false) {
        if ($subDirName == '.' || $subDirName == '..') {
            continue;
        }
        //Путь до подкатегорий с файлами
        $subDirPath = "$dirPath/$subDirName";
        //Счётчик файдов в директории
        $filesCount = 0;
        $rootSubDir = opendir($subDirPath);
        while (($fileName = readdir($rootSubDir)) !== false) {
            if ($fileName == '.' || $fileName == '..') {
                continue;
            }
            if (in_array($fileName, $arFiles)) {
                $filesCount++;
                continue;
            }
            //Полный путь до файла
            $fullPath = "$subDirPath/$fileName";
            //Удаление файла
            unlink($fullPath);
        }
        closedir($rootSubDir);
        //Удалить подкатегорию, если она пуста
        if (!$filesCount) {
            rmdir($subDirPath);
        }
    }
}
```

```

    }
}
closedir($rootDir);
return __METHOD__."()";
}

```

2) В файл `/bitrix/php_interface/init.php` добавляем строки:

```

<?php
if (file_exists($_SERVER["DOCUMENT_ROOT"]."/bitrix/php_interface/include/agents.php"))
    require_once($_SERVER["DOCUMENT_ROOT"]."/bitrix/php_interface/include/agents.php");

```

3) Настроим выполнение агента. Заходим Настройки > Настройки продукта > Агенты и заполняем форму следующим образом:

Задача №2

написать скрипт который будет добавлять товар и его торговые предложения
Задача должна быть реализована как отдельный скрипт с использованием Bitrix API

Решение:

Можно написать миграцию, которая добавит товар и его предложение (`/bitrix/php_interface/migrations/Version20190714.php`).

```

<?php
namespace Sprint\Migration;
use \Bitrix\Main\Loader;
use Exception;
class Version20190714 extends Version {
    protected $description = "Добавить товар и его торговые предложения";
    function up()
    {
        // TODO: Implement up() method.
        $this->out("Начало ...");
        if (!Loader::includeModule("iblock")) {
            $this->outError("Не удалось запустить модуль [Информационные блоки]");

```

```

        return;
    }
    if (!Loader::includeModule("catalog")) {
        $this->outError("Не удалось запустить модуль [Каталог]");
        return;
    }
    $catalogOffersIblockId = 2; // Ид инфоблока предложений
    $productName = "Диван"; // Наименование товара
    $offerName = "Диван чёрный цвет"; // Наименование торгового предложения
    $offerPrice = 45000; // Цена торгового предложения
    $arCatalog = \Bitrix\Catalog\CatalogIblockTable::getRow([
        'filter' => [
            '=IBLOCK_ID' => $catalogOffersIblockId
        ],
        'select' => ['ID', 'PRODUCT_IBLOCK_ID', 'SKU_PROPERTY_ID']
    ]);
    $iblockCatalogId = $arCatalog["PRODUCT_IBLOCK_ID"]; // Ид инфоблока товаров
    $skuPropertyId = $arCatalog["SKU_PROPERTY_ID"]; // ID свойства в инфоблоке
    предложений типа "Привязка к товарам (SKU)"
    $arFields = [
        'NAME' => $productName,
        'IBLOCK_ID' => $iblockCatalogId,
        'ACTIVE' => 'Y'
    ];
    $objElement = new \CIBlockElement();
    $productId = $objElement->Add($arFields); // Добавить товар
    if ($productId) {
        $objElement = new \CIBlockElement();
        // Свойство торгового предложения
        $arOfferProps = [
            $skuPropertyId => $productId
        ];
        $arOfferFields = [
            'NAME' => $offerName,
            'IBLOCK_ID' => $catalogOffersIblockId,
            'ACTIVE' => 'Y',
            'PROPERTY_VALUES' => $arOfferProps
        ];
        // Добавить торговое предложение
        $offerId = $objElement->Add($arOfferFields);
        if ($offerId) {
            // Добавить как товар
            try {
                \Bitrix\Catalog\ProductTable::add([
                    'ID' => $offerId,
                    'VAT_INCLUDED' => "Y" // НДС входит в стоимость
                ]);
            } catch (Exception $e) {
                $this->outError($e->getMessage());
            }
            try {
                // Указывать цену
                \CPrice::SetBasePrice($offerId, $offerPrice, "RUB");
            } catch (Exception $e) {
                $this->outError($e->getMessage());
            }
        } else {
            $this->outError("Ошибка добавления торгового предложения: " .
                $objElement->LAST_ERROR);
            return;
        }
    } else {
        $this->outError("Ошибка добавления товара: " . $objElement->LAST_ERROR);
        return;
    }
}

```

```

    }
}
function down()
{
    // TODO: Implement down() method.
}
}

```

Задача №3

<https://docs.google.com/spreadsheets/d/1AuTBQzV7P1c82XhJkynuQj4-xbKWhlGJ5J3BkG40U4w/edit?usp=drivesdk>

Нужно взять chart.json и написать скрипт который на выходе будет отдавать chart_result.json

<https://drive.google.com/drive/folders/16tDtdgO9tw41ORuhO6tVazPUKS7TxeDP>
(тут файлы)

Решение:

fill_null.php

```

<?php
// Имя читаемого файла
$fileName = "chart.json";
// Данные из файла
$fileContent = @file_get_contents($fileName);
// Преобразование в массив
$chartList = @json_decode($fileContent, true);
// Счетчик количество значение 100 для каждого столбца
$cnt_100 = [];
for ($i = 0; $i < count($chartList); $i++) {
    for ($j = 1; $j < count($chartList[$i]); $j++) { // Начинаем с 1-ого элемента, 0-ое
        значение пропустим (time)
        if ($chartList[$i][$j] == 100) { // Посчитаем длины последовательности 100
            $cnt_100[$j]++;
        } else { // Если отличное от 100
            $cnt_100[$j] = 0;
        }
        // Если длина последовательности 100 больше 3, то меняем значение на null
        if ($cnt_100[$j] == 4) {
            for ($k = 0; $k <= 3; $k++) {
                $chartList[$i - $k][$j] = null;
            }
        } elseif ($cnt_100[$j] > 4) {
            $chartList[$i][$j] = null;
        }
    }
}
// Имя файла для вывода
$resultFileName = "chart_result.json";
// Написать данные в файл
file_put_contents($resultFileName, json_encode($chartList));

```