


```
class User implements UserLanguagePreferences
{
    private $storage;

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

    public function setLanguage(string $lang)
    {
        $this->storage->set(UserLanguagePreferences::STORAGE_KEY, $lang);
    }

    public function getLanguage():? string
    {
        return $this->storage->get(UserLanguagePreferences::STORAGE_KEY);
    }
}

$storage = new SessionStorage();
$user = new User($storage);
```























































```
class User implements UserLanguagePreferences
{
    private $storage;

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

    public function setLanguage(string $lang)
    {
        $this->storage->set(UserLanguagePreferences::STORAGE_KEY, $lang);
    }

    public function getLanguage():? string
    {
        return $this->storage->get(UserLanguagePreferences::STORAGE_KEY);
    }
}

$storage = new SessionStorage();
$user = new User($storage);
```

← ОЧЕНЬ ЛЕГКО КАСТОМИЗИРОВАТЬ

```
class MysqlStorage implements StorageInterface
{
    private $mysqlStorage;

    public function __construct()
    {
        // Some mysql engine init code
    }

    public function set(string $key, string $value): StorageInterface
    {
        $this->mysqlStorage->set($key, $value);
    }

    public function get(string $key):? string
    {
        return $this->mysqlStorage->get($key);
    }
}

$storage = new MysqlStorage();
$user = new User($storage);
```


