

відобразатиметься базова інформація про роботи. Іншою вимогою є забезпечення детального перегляду кожного наряду де відобразатиметься повна та детальна інформація. Також необхідно передбачити відображення активних нарядів у режимі реального часу та відображення історії скасованих та завершених нарядів використовуючи сторінкову навігацію. Окрім того, додаток повинен містити можливість фільтрації нарядів за основними інформативними полями. Редактор нарядів повинен складатись з декількох вкладок - для кожного виду робіт: основної механізованої, допоміжної механізованої та допоміжної ручної. Віконні форми розробленого додатку обліку аграрного підприємства наведено на рис. 3.

Висновок

В роботі проведено опис критеріїв, яким повинно задовольняти програмне забезпечення обліку робіт аграрного підприємства. Здійснено визначення предметних підобластей та взаємозв'язків між ними на основі яких розроблено архітектуру програмного забезпечення обліку робіт аграрного підприємства. Використовуючи спроектовану архітектуру було розроблено програмне забезпечення обліку робіт аграрного підприємства.

Література

1. Fowler, M., Rice D., Foemmel, M., Hieatt, E., Mee, Stafford, R.: Patterns of Enterprise Application Architecture, Addison-Wesley, pp. 533 (2014).
2. Gramma, E., Helm, R., Johnson, R., Vlissides J.: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, pp. 403 (2000).
3. Richter, J.: CLR via C# (Developer Reference), 4th Edition, Microsoft Press, pp. 896-902 (2012).
4. Garlan, D., Shaw, M.: An Introduction to Software Architecture, Technical Report, University Pittsburgh, PA, USA, pp. 457 (1994).
5. Clements, P.C., Heitmeyer, C.L., Labaw, B.G., Rose, A.T.: MT: A toolset for specifying and analyzing real-time systems, In Proc. of the Real-Time Systems Symposium 93, pp. 12-22 (1993).

Отримана/Received : 15.5.2017 р. Надрукована/Printed : 10.6.2017 р.
Рецензент: д.т.н., проф. Боровик О.В.

УДК 004.415

О.А. КРАВЧУК

Хмельницький національний університет

ПРАКТИЧНИЙ ОГЛЯД PHP 7

В статті розглядається PHP7 – мова програмування загального призначення з відкритим вихідним кодом. PHP сконструйований спеціально для ведення веб-розробок. PHP – це кросплатформенна технологія. Дистрибутив PHP доступний для більшості операційних систем, включаючи Linux, багатьох модифікацій Unix, Microsoft Windows, Mac OS. PHP підтримується на більшості веб-серверів, таких як Apache, Nginx, Microsoft Internet Information Server (IIS), Microsoft Personal Web Server та інших.

Ключові слова: PHP7, мова програмування, 7 версія, висока швидкодія, оптимізація, мультиплексування, веб-сервер, відкритий код, веб-розробка.

О.А. KRAVCHUK
Khmelnytsky National University

PRACTICAL PHP 7 REVIEW

In the article we review PHP7 - general-purpose programming language. It is open source software. PHP is designed specifically for web development. PHP it is cross-platform technology. Distribution PHP is available for most operating systems, including Linux, many modifications of Unix, Microsoft Windows, Mac OS. PHP is supported on most web servers such as Apache, Nginx, Microsoft Internet Information Server (IIS), Microsoft Personal Web Server, and many others. The most easily recognizable advantage of the new PHPNG engine is the significant performance improvement. The development team of PHPNG refactored the Zend Engine, and remarkably optimized memory usage. By using PHP 7 not only your code will be executed faster but you will also need fewer servers to serve the same amount of users.

Keywords: PHP7, programming language, version 7, high performance, optimization, multiplexing, Web server, open source, web development.

Постановка проблеми. Однією з найбільш значних подій, що відбулися в світі програмування, став випуск PHP 7. Цілих 10 років відділяють його від випуску першого релізу PHP з номером 5. Зі збільшенням першої цифри в номері версії, в PHP 7 з'явилася маса нововведень, збільшилася й швидкість роботи.

Практично всі оновлення додавали невеликі поліпшення продуктивності PHP. Однак на цей раз продуктивність PHP, в порівнянні з ранніми версіями, зросла куди значно більше, ставши однією з найочікуваніших особливостей PHP 7. Це було частиною проекту "PHPNG" ("php new generation"), яка

зачепила власне сам Zend Engine [2].

Рефакторинг внутрішніх структур даних і додавання додаткового етапу перед компіляцією коду у вигляді абстрактного синтаксичного дерева – Abstract Syntax Tree (AST) привели до надзвичайної продуктивності та більш ефективному розподілу пам'яті. Цифри самі по собі виглядають багатообіцяюче – тести, виконані на реальних додатках показують, що PHP 7 в середньому вдвічі швидше PHP 5.6, а також використовує на 50% менше пам'яті під час обробки запитів, що робить PHP 7 сильним суперником для компілятора HHVM JIT від Facebook.

В сьомій версії є лише один недолік, оскільки була прибрана застаріла функціональність, це призвело до деяких проблем з зворотною сумісністю, ускладнивши переведення старих додатків на нову версію.

Аналіз досліджень. Якщо ви якийсь час не працювали з PHP, ви можете здивуватися, куди ж пропав PHP 6, чому відбувся стрибок з PHP 5 відразу на PHP 7? Випуск PHP шостої версії не відбувся. Головною особливістю версії 6 була підтримка символів Unicode, оскільки в основному PHP використовується в веб-розробці, вебу ж потрібна підтримка Unicode, тому її реалізація мала сенс [3].

Ідея була в тому, щоб реалізувати підтримку Unicode безпосередньо в ядрі. Такий підхід мав розширити можливості мови – від використання неоднозначних емоджі, як імена змінних і функцій, до потужної функціональності інтернаціональних рядків. На жаль, цей амбітний план мав куди більше проблем, ніж очікувалося. Велика частина кодової бази повинна була бути перенесена для підтримки Unicode, як у випадку ядра, так і в разі важливих розширень, що виявилось дуже важкою і непростою справою. Це уповільнило процес розробки інших можливостей мови, що засмутило багатьох PHP-розробників. З'явилися і інші бар'єри, що призвело до падіння інтересу до розробки вбудованої підтримки Unicode, а згодом проект і зовсім виявився анти-продуктивним [4]. А оскільки маса ресурсів, таких як книги і статті, були написані з використанням назви PHP 6 новій версії було вирішено присвоїти ім'я PHP 7 – просто для того, щоб уникнути непорозумінь.

Формулювання цілей статті. Мета дослідження – огляд та аналіз нововведень мови програмування PHP7 та проблеми, які можуть виникнути при перенесенні веб-додатків на дану версію.

Виклад основного матеріалу дослідження. PHP – це мова програмування, яка в основному застосовується в веб-розробці, як правило, в частині back-end. PHP має досить довгу історію – його перша версія з'явилася більше 20 років тому, в 1995 році. За цей час було випущено 7 версій PHP, остання з яких з'явилася в грудні 2015 року. Про неї і піде мова в цій статті.

Реліз PHP 7.0 став у своєму роді проривом, оскільки він ґрунтувався на phpng – експериментальному і активно прогресуючому проекті PHP, назва якого розшифровується як "PHP Next Generation" – «Наступне покоління PHP». Дану технологію можна було впровадити і в попередніх версіях PHP, проте саме в PHP 7 вона була введена як осново-утворююча. При її розробці була поставлена мета підвищити продуктивність PHP, і при цьому не втратити сумісність.

PHP 7 заснований на третій версії Zend Engine, в якій був розвинений проект phpng. В першу чергу, нововведення в PHP 7 торкнулися найважливіших для розробки додатків параметрів – це продуктивність і споживання пам'яті.

Основна зміна, про який в першу чергу згадують в розмові про PHP 7, це поліпшення продуктивності: показники PHP 7 наближені до показників HHVM (розробленої Facebook системи, яка дозволяє прискорити перетворення PHP-коду в кілька разів). Не останньою причиною цього стало те, що співзасновник Zend Technologies (компанії-розробника ядра PHP) Зеєв Сураські розглядає HHVM як одного з конкурентів Zend. А провідний розробник PHP 7 компанії Zend Technologies і автор phpng Дмитро Стогів запевняє, що різниця в продуктивності PHP 5.0 і PHP 7 на синтетичному тесті Bench PHP становить 14 раз. На практиці більшість користувачів схиляються до думки, що PHP 7 приблизно в два рази швидше попередньої версії.

HHVM використовує так звану JIT (Just-In-Time) компіляцію для перетворення коду – незважаючи на те, що JIT в 7 версії PHP не з'явився, PHP 7 підтримує більш хорошу міграцію, яка і дозволяє поліпшити продуктивність в рази. Саме новий рівень продуктивності Дмитро Стогів виділяє в якості головної особливості PHP 7.

Поліпшення продуктивності вдалося домогтися і завдяки тому, що компілятор тепер генерує більше ефективний і короткий байт-код.

В наш час вкрай важливо думати про користувачів не тільки стаціонарних, але і мобільних пристроїв, тому розробники PHP 7 поліпшили виконання операцій ядра і знизили споживання пам'яті, а також додали можливість використовувати нативне локальне сховище. До того ж PHP 7 дозволяє виконувати асинхронні завдання. До речі, при розробці враховувалися і популярні тенденції щодо скорочення споживання ресурсів планети, тому PHP 7 має можливість обробляти більше трафіку, використовуючи все ті ж ресурси сервера.

Розглянемо основні нововведення сьомої версії. У PHP 7 з'явилося кілька нових синтаксичних особливостей, одним з них є групування оголошень імпорту. Тепер синтаксис мови дозволяє групувати оголошення імпорту класів, які перебувають в одному просторі імен, в одному рядку. Це допоможе нам вирівняти декларації якимось, наділеним сенсом, чином або ж просто заощадить пару байт вашого коду.

Наприклад:

```
use Framework \ Module \ Foo;
use Framework \ Module \ Bar;
use Framework \ Module \ Baz;
```

У PHP 7 можна написати:

```
use Framework \ Module \ {Foo, Bar, Baz};
```

Або ж, якщо ви віддаєте перевагу багатостроковому стилю:

```
use Framework \ Module {
    Foo,
    Bar,
    Baz
};
```

Наступним нововведенням є Null-коалесцентний оператор. Він вирішує поширену проблему в PHP. Вона виникає в разі, якщо ми хочемо мати значення змінної, яке присвоєно іншій змінній, але якщо останній змінній значення не було присвоєно, то привласнити якимось явним значенням. Часто виявляється необхідна при роботі з призначенням для користувача вводом даних.

Нижче наведено приклад виникнення такої ситуації:

До PHP 7:

```
if (isset ($ foo)) {
    $ Bar = $ foo;
} Else {
    $ Bar = 'default'; // Присвоюємо $ bar значення 'default' якщо $ foo дорівнює NULL
}
```

У PHP 7 дана конструкція має наступний вигляд:

```
$ Bar = $ foo ?? 'Default';
```

Цей вираз можна використовувати з ланцюжком змінних:

```
$ Bar = $ foo ?? $ Baz ?? 'Default';
```

Також новим в синтаксисі є оператор `<=>`. Він дозволяє проводити тривірне порівняння двох значень, дозволяючи розуміти не тільки їх рівність чи нерівність, а й те, яке з них більше при нерівності, повертаючи 1,0 або -1.

У цьому випадку ми можемо робити різні дії в залежності від того, як розрізняються значення, нижче наведений приклад коду:

```
switch ($ bar <=> $ foo) {
    case 0:
        echo '$ bar і $ foo рівні';
    case -1:
        echo '$ foo більше';
    case 1:
        echo '$ bar більше';
}
```

Порівнювані значення можуть мати тип `integer`, `float`, `string` і навіть бути масивами.

У PHP 7 розширили раніше існуюче оголошення параметрів в методах (класах, інтерфейсах і масивах) шляхом додавання чотирьох скалярних типів – цілого (`int`), з плаваючою комою (`float`), логічного (`bool`) і строкового (`string`) в якості можливого типу параметра.

Крім того, опціонально ми можемо вказати тип результату, що повертається функцією або методом.

Підтримуються типи `bool`, `int`, `float`, `string`, `array`, `callable`, ім'я класу або інтерфейсу і `parent` (для методів класу).

Наведемо приклад для цього нововведення:

```
class Calculator
{
    // Оголошуємо, що параметри мають цілий тип integer
    public function addTwoInts (int $ x, int $ y): int {
    // Явно оголошуємо, що метод повертає ціле
        return $ x + $ y;
    }
}
```

Оголошення типів дозволить будувати прозорі виклики, уникаючи передачі та повернення невірних значень при роботі з функціями. Інші плюси – поява аналізаторів статичного коду і IDE, що пропонують більш чітке відображення коду при відсутності документуючих приміток DocBlocks.

Оскільки PHP слабо типізована мова, деякі значення параметрів і типів будуть приводитися виходячи з контексту. Якщо ми передаємо значення "3" в функцію, що має оголошений параметр типу `int`, інтерпретатор розглядатиме його як ціле і не згенерує помилку. Якщо вас не влаштовує така поведінка, ви можете працювати в суворому режимі – `strict mode` – завдяки додаванню відповідної директиви в параметрах середовища.

```
declare (strict_types = 1);
```

Дана опція встановлюється окремо для кожного файлу, оскільки глобальна опція розділила б сховища коду на такі, в яких файли розроблялися з включеною опцією і ті, де вона була б відключена, що

привело б до непередбачуваної поведінки при роботі з кодом з двох сховищ з різними налаштуваннями.

Наступним важливим нововведенням PHP 7 є обробка винятків рівня ядра.

З появою винятків рівня ядра, фатальні помилки, які раніше могли призвести до зупинки виконання коду, тепер можуть бути легко перехоплені і оброблені.

Помилки, такі як виклик неіснуючого методу, тепер не зупинять скрипт, замість цього буде згенеровано виняток, який можна обробити в блоці try catch, що явно покращує обробку помилок у вашому додатку. Це важливо для деяких типів додатків, серверів і демонів, оскільки фатальні помилки, в іншому випадку, цілком могли привести до необхідності їх рестарту. Тести в PHPUnit також повинні стати більш зручними у використанні, оскільки фатальні помилки могли зупинити весь тестувальний проект. Винятки, на відміну від помилок, можуть бути оброблені для кожного тесту окремо.

У PHP 7 з'явилося досить багато нових класів винятків, покликаних обробляти типи помилок, з якими ви можете зіткнутися. Для забезпечення сумісності між версіями, доданий новий інтерфейс Throwable, він може бути реалізований як винятками рівня ядра, так і користувацькими винятками. Такий підхід реалізований з метою недопущення спадкування базового класу винятків винятками ядра, що призвело б до появи винятків в попередньо написаному коді, яких раніше не було. Наведемо приклад.

До PHP 7 такий код привів би до фатальної помилки виконання:

```
try {
    thisFunctionDoesNotExist (); // ЭтаФункцияДажеНеСуществует ()
} catch (\ EngineException $ e) {
    // Підчищаємо за собою і записуємо інформацію про помилку в лог
    echo $ e-> getMessage ();
}
```

Наступне нововведення в PHP 7 – це анонімні класи. Анонімні класи – двоюрідні брати анонімних функцій, які ви могли використовувати при створенні функціональності коротких об'єктів. Анонімні класи легко створюються і використовуються так само, як і звичайні об'єкти. Ось приклад з документації:

До PHP 7:

```
class MyLogger {
    public function log ($ msg) {
        print_r ($ msg. "\ n");
    }
}

$ Pusher-> setLogger (new MyLogger ());
```

Використання анонімного класу:

```
$ Pusher-> setLogger (new class {
    public function log ($ msg) {
        print_r ($ msg. "\ n");
    }
});
```

Анонімні класи корисні при тестуванні юнітами, зокрема при мокінгу (імітації поведінки реального об'єкта) при тестуванні об'єктів і сервісів. Їх наявність дозволить нам уникнути використання великих мокінг-бібліотек і фреймворків шляхом створення простого об'єкта, що підтримує інтерфейс, який ми можемо використовувати для мокінгу.

Нововведення торкнулися також і блоку криптографічних функцій, а саме це дві нових функції для генерації криптографічно безпечних рядків і цілих. Перша повертає випадковий рядок довжиною \$ len:

```
random_bytes (int $ len);
```

Друга повертає число в діапазоні \$ min ... \$ max.

```
random_int (int $ min, int $ max);
```

Дані функції дозволять зробити більш безпечною роботу при авторизації в додатках. Також оновлені генератори. Генератори в PHP теж отримали кілька приемних нових можливостей. Тепер у них з'явився оператор return, який може бути використаний для видачі деякого фінального значення, актуального на момент завершення ітерації. Його можна використовувати для перевірки коректності виконання генератора. Наприклад, дізнатися, виконався він без помилок, що дозволить коду, який викликав генератор, коректно обробити будь-яку ситуацію, що виникла.

Більше того, генератори можуть повертати і видавати вирази з інших генераторів. Таким чином можна розбивати складні операції на більш прості, наведемо відповідні приклади:

```
function genA () {
    yield 2;
    yield 3;
    yield 4;
}

function genB () {
    yield 1;
```

```

yield from genA (); // 'GenA' викликається і відпрацьовує в цьому місці
yield 5;
return 'success'; // Фінальний результат, який ми пізніше можемо перевірити
}

foreach (genB () as $ val) {
    echo "\ n $ val"; // Видасть значення від 1 до 5
}

$ GenB () -> getReturn (); // Поверне 'success' при відсутності помилок

```

В 7 версії PHP також доданий синтаксис Escape-коду для Unicode. На відміну від багатьох інших мов, до PHP версії 7, в PHP не було способу вказати в рядку escape-послідовність для Unicode символу. Тепер за допомогою escape-послідовності `\u` можна генерувати такі символи за допомогою їх кодів з набору UTF-8. Це краще, ніж безпосередня вставка символів, краще контролюються невидимі символи і символи, які мають графічне відображення відмінне від значення, приклад:

```
echo "\ u {1F602}"; // Виводить смайлик
```

До нововведень також відносяться «Очікування» (expectations) – поліпшення функції `assert ()` зі збереженням зворотної сумісності. Вони дозволяють використовувати затвердження з нульовою вартістю (zero-cost assertions) в робочому коді і підтримують можливість генерації користувацького виключення при виникненні помилки при відпрацюванні твердження, що може бути корисно при розробці.

Функція `assert ()` стала мовною конструкцією в PHP 7. Твердження повинні бути використані тільки під час розробки та тестування з метою налагодження. Для настройки її поведінки ми повинні використовувати дві директиви.

```
zend.assertions
```

1: генеруємо і виконуємо код (режим розробки) (значення за замовчуванням);

0: генерує код, але обходить його під час виконання;

-1: Не генерує код, роблячи його кодом з нульовою вартістю (режим робочого коду).

```
assert.exception
```

1: генерується при помилці затвердження шляхом створення відповідного об'єкта виключення або ж шляхом генерації об'єкта `AssertionError`, якщо такий об'єкт не був створений

0: використовує або генерує `Throwable` так, але генерується тільки попередження (warning) на базі того об'єкта, а не генерація виключення з його допомогою.

Готуємося до переходу від PHP 5 до PHP 7.

Поява версії PHP з номером 7 надало можливість зміни / оновлення і навіть видалення функціональності, яка вважалася застарілою або, деякий час, вже непотрібною. Такі зміни можуть призвести до проблем зі зворотним сумісністю старих додатків.

Інша проблема, що виникає при появі подібних версій, полягає в тому, що бібліотеки і фреймворки, які для вас важливі, можуть не мати оновлень, сумісних з новим релізом. Команда розробників PHP намагалася вносити зміни якомога більш високого ступеня зберігши зворотну сумісність для того, щоб міграція на нову версію була якомога менш трудомісткою. Відносно нові і вчасно оновлювані програми, швидше за все, буде простіше перевести на сьому версію, тоді як для старих додатків необхідно зважене рішення – чи варті витрачені зусилля отриманих можливостей або ж краще і зовсім не оновлюватися, значна частина проблем невеликих об'єктів і можуть бути легко усунені, в той час як інші можуть зажадати куди більше часу і зусиль. В цілому, якщо ви бачили попередження про використання застарілих функцій до установки PHP 7, то ви, швидше за все, побачите повідомлення про помилки, які припинять виконання вашої програми до їх усунення.

Куди важливішим є той факт, що були прибрані старі і непотрібні SAPI, наприклад такі як розширення `mysql`. Повний список розширень і можливостей, які були видалені, ви можете подивитися тут і тут.

Інші розширення і SAPI були перенести на PHP 7.

Багато сервісів хостингу додали підтримку PHP 7. Це хороша звістка для провайдерів `shared-хостингу`, оскільки збільшення продуктивності дозволить їм збільшити число клієнтських веб-сайтів без поновлення заліза, зменшивши поточні витрати і збільшивши доходи. Що до клієнтів, то вони не повинні чекати різкого збільшення продуктивності в цих умовах, та й, відверто кажучи, `shared-хостинг` в будь-якому разі не опція в разі застосування, орієнтованого на продуктивність.

З іншого боку, сервіси, які пропонують приватні віртуальні сервера або виділені сервери, отримують всю красу зростання продуктивності. Деякі PaaS-сервіси, такі як `Heroku`, давно підтримують PHP 7, інші ж, типу `AWS Beanstalk` і `Oracle OpenShift`, плетуться позаду. Переконайтеся на веб-сайті вашого PaaS-провайдера, чи є у нього підтримка PHP 7 або ж вона планується в найближчому майбутньому.

На додачу до сумісності з інфраструктурою, вам також необхідно пам'ятати про потенційні проблеми сумісності з програмним забезпеченням. Відомі CMS типу `WordPress`, `Joomla` і `Drupal` вже додали підтримку PHP 7. Основні фреймворки, такі як `Symfony` і `Laravel` теж зробили це.

Однак настав час застережень. Ця підтримка не поширюється на код третіх осіб у вигляді доповнень, плагінів, пакетів і т.п., до чого звертається ваш CMS або фреймворк. Проблеми можуть бути, і ваше завдання полягає в тому, щоб переконатися – все готово для роботи під PHP 7.

Висновки. Таким чином, у PHP 7 був прибраний застарілий код, а також було дано «зелене світло» новим можливостям і майбутнім поліпшенням в області ефективності. Плюс, PHP незабаром повинен отримати оптимізацію продуктивності. Незважаючи на часткову втрату зворотної сумісності з попередніми версіями, більшість виникаючих проблем легко вирішувати.

Бібліотеки та фреймворки мігрують на PHP 7, що призводить до появи їх нових версій. Я закликаю вас спробувати сьому версію і оцінити отримані результати. Може бути, ваш додаток вже сумісний з новою версією і готовий працювати на PHP 7, отримуючи вигоди від його використання.

Література

1. Пелешин А.М. Позиціонування сайтів у глобальному інформаційному середовищі / А.М. Пелешин. – Львів : Вид-во Національного університету "Львівська політехніка", 2007. – 258 с.
2. Steve Prettyman. Learn PHP 7: Object Oriented Modular Programming using HTML5, CSS3, JavaScript, XML, JSON, and MySQL. 2016. 308 p.
3. Antonio Lopez. Learning PHP 7. 2016. 414 p.
4. Mikael Olsson. PHP 7 Quick Scripting Reference, Second Edition. 2016. 136 p.

Отримана/Received : 24.4.2017 р. Надрукована/Printed : 11.6.2017 р.

Рецензент: д.ф.м.н., проф.Бедратюк Л.П.

УДК 004.056

О.П. ВОЙТОВИЧ, М.В. ГУРСЬКИЙ, Л.М. КУПЕРШТЕЙН

Вінницький національний технічний університет

Д.С. СНИГОВИЙ

Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"

ЗАСІБ МОНІТОРИНГУ ДЛЯ ОПЕРАЦІЙНОЇ СИСТЕМИ ANDROID

На основі аналізу основних методів та шляхів поширення шкідливого програмного забезпечення в ОС Android, розроблено програмний засіб для моніторингу вхідного/вихідного трафіку усіх системних та встановлених програмних застосунків.

Ключові слова: операційна система Android, загрози та вразливості, інформаційна безпека, система моніторингу, шкідливе програмне забезпечення.

O.P. VOITOVYCH, M.V. HURSKYI, D.S. SNIGOVYY

Vinnytsya National Technical University

L.M. KUPERSHTEIN

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

MONITORING TOOL FOR ANDROID OPERATING SYSTEM

A purpose of this article is the improvement of information security of mobile devices using analysis of main threats, their ways of penetrating the system using monitoring of installed applications. Due to the high popularity of Android devices where getting root privilege's is very simple the problem of installing software from untrusted sources and increasing the likelihood of inadvertent introduction of malicious code into mobile device is actual as never before. The problem is getting even worse with that fact that a big part of used devices are personal which means that user can use them as he want, for example, connect to production servers with trade secrets, accounting, bank accounts, etc. In this article architecture and main problems of protecting Android operation systems such as types of malware and methods of their spreading were reviewed. Based on detected ways a policy that will minimize the likelihood of malware to device and save private data from leaks was developed. One of the methods is to use monitoring software. Using it user can analyze all of the systems and minimize risks. Developed software tool allows real-time monitoring of the system and all installed applications connections to the Internet.

Keywords: operating system Android, threat and vulnerability, information security, monitoring system, malware.

Вступ

На сьогоднішній день ринок мобільних пристроїв вже обігнав ринок персональних комп'ютерів. В той же час стрімке зростання обчислювальної потужності і можливостей мобільних пристроїв ставлять нові питання і проблеми в галузі забезпечення інформаційної безпеки. Поширення мобільних пристроїв тягне за собою зростання бажання заволодіти як фізично цими самими пристроями, так і інформацією, яка зберігається на них. Найбільше хакерів приваблює операційна система Android в силу своєї відкритості та широкої розповсюженості [1]. Обсяги шкідливого коду ростуть з роками майже в геометричній прогресії. У зв'язку з високою популярністю пристроїв Android, в якому отримання root-прав робиться в пару дотиків, проблема встановлення додатків з недовірених джерел, і, як наслідок, збільшення ймовірності ненавмисного впровадження шкідливого коду на мобільний пристрій, актуальна як ніколи [2].

У своїх дослідженнях В. В. Казимир та І. І. Карпачев [3] довели, що, на практиці, антивірусний захист надає обмежену безпеку, або навіть її ілюзію, а Saad M. H., Serageldin A., Salama G. [4] у своїй праці довели, що лише користувач є винним у несанкціонованому доступі до власних даних, винятком є лише вразливості у оновленнях ОС. На сьогоднішній день існує багато програмних та організаційних методів, в