

Національний технічний університет України
«Київський політехнічний інститут»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Розрахунково-графічна робота
по курсу
«Інтеграційні програмні системи »

Виконали: студенти 4курсу

ФІОТ гр. ІО-44

Бригада кр1 в складі:

Лопатін Євген

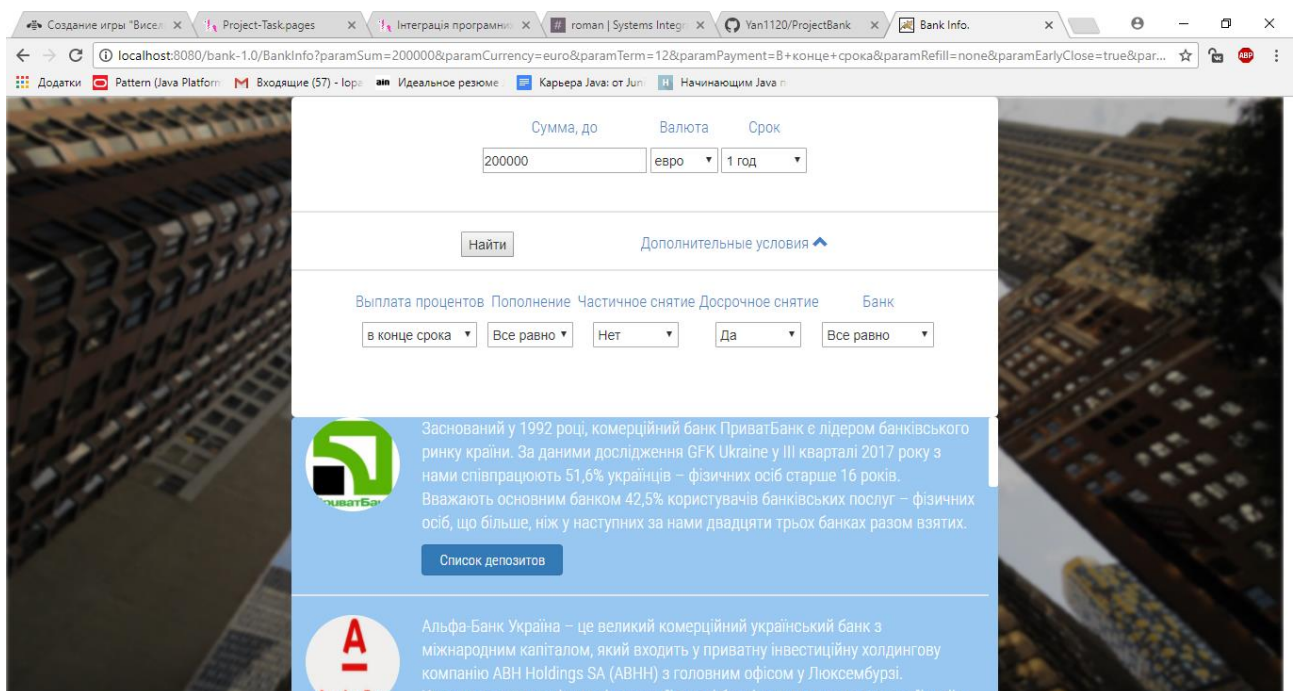
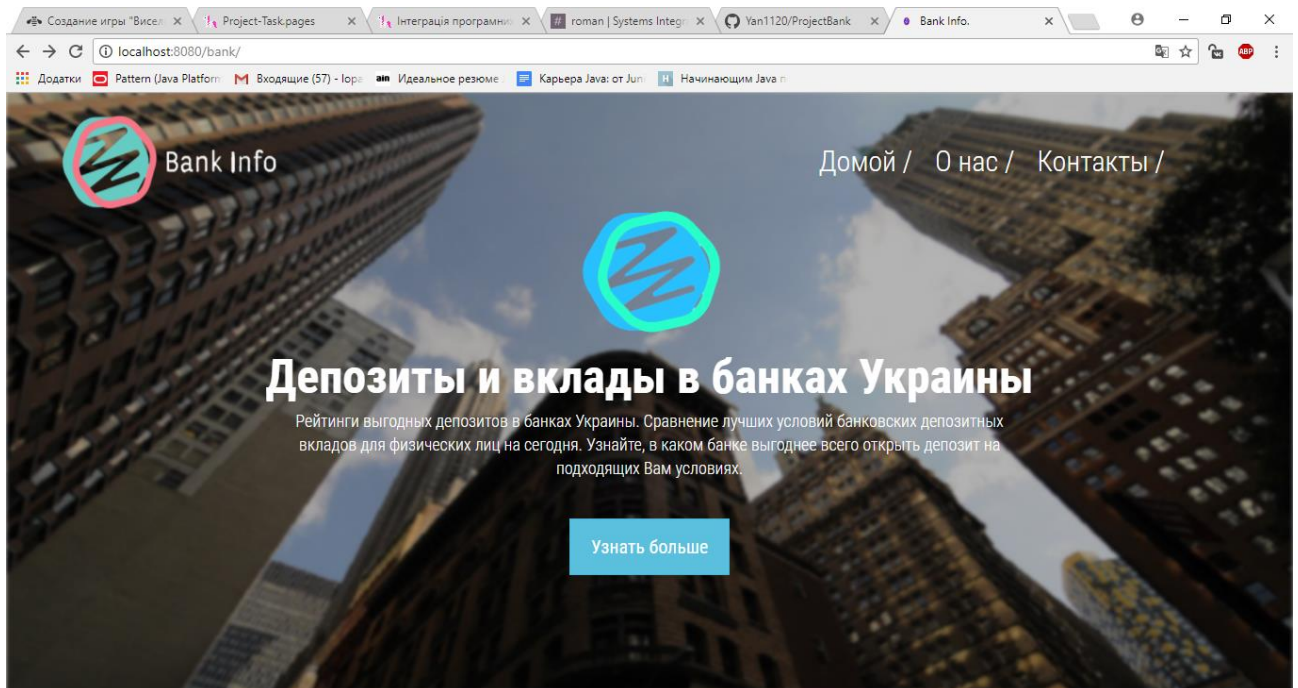
Пелих Олександр

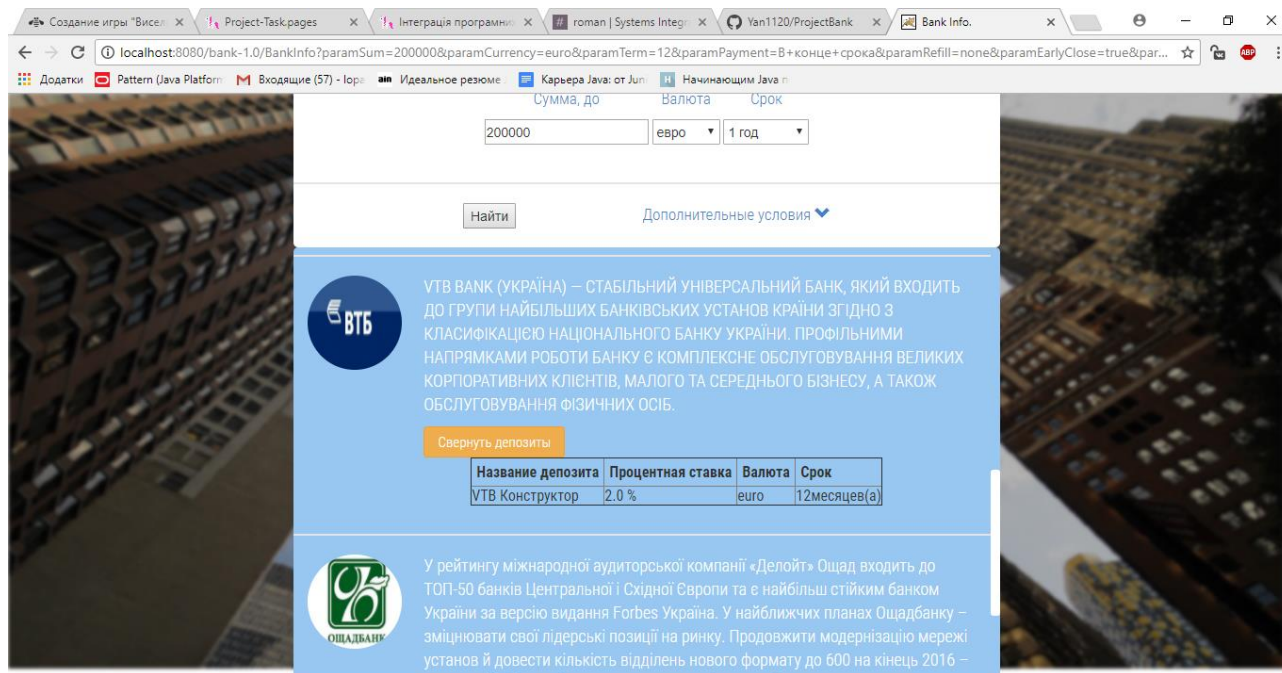
Калітинський Ярослав

Київ 2017

1. Опис проекту

Розроблений нами проект дозволяє користувачам переглядати депозити, які Українські банки надають своїм клієнтам, вибирати умови депозиту які потрібні саме клієнту наприклад валюту, термін тощо. і після вибору депозиту перенаправлятись на сайт банку цього депозиту (дана функція знаходиться в процесі розробки).





Скріншот роботи сервісу

Серверна частина сервісу розроблена за допомогою засобів Java EE. В основі архітектури лежить паттерн MVC, який реалізовано за допомогою Сервлетів, JSP, jstl та Tomcat. Дизайн включає в себе HTML, CSS і Javascript

Бібліотека тестування Junit

2. Система атоматичної збірки. Maven

Maven - це засіб автоматизації роботи з програмними проектами, який спочатку використовувався для Java проектів. Використовується для управління (management) та складання (build) програм. Створений Джейсоном ван Зилом (*Jason van Zyl*) у 2002 році. За принципами роботи кардинально відрізняється від Apache Ant, та має простіший вигляд щодо build-налаштувань, яке надається в форматі XML. XML-файл описує проект, його зв'язки з зовнішніми модулями і компонентами, порядок будування (build), папки та необхідні плагіни. Сервер із додатковими модулями та додатковими бібліотеками розміщується на серверах. Раніше Maven, де він був частиною *Jakarta Project*.

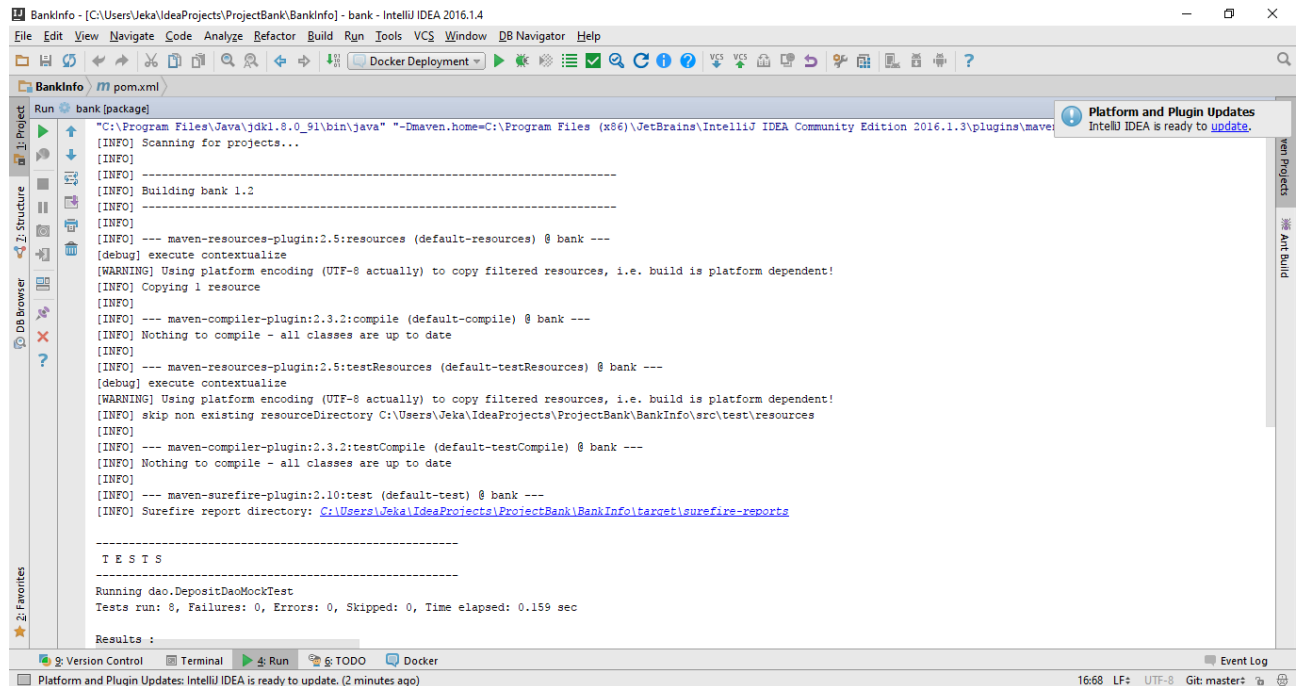
Для опису програмного проекту який потрібно побудувати (*build*), Maven використовує конструкцію відому як Project Object Model (POM), залежності від зовнішніх модулів, компонентів та порядку побудови. Виконання певних, чітко визначених задач - таких, як компіляція коду та пакетування відбувається шляхом досягнення заздалегідь визначених цілей (targets).

Ключовою особливістю Maven є його мережева готовність (network-ready).

Двигун ядра може динамічно завантажувати плагіни з репозиторію, того самого репозиторію, що забезпечує доступ до багатьох версій різних Java-проектів з відкритим кодом, від Apache та інших організацій та окремих розробників. Цей репозиторій та його реорганізований наступник, - Maven 2 репозиторій, - намагається бути де-факто механізмом для дистрибуції Java програм, але прийняття його в такій якості йде повільно.

Maven забезпечує підтримку побудови не просто перебираючи файли з цього репозиторію, але й завантажуючи назад артефакти у кінці побудови. Локальний кеш звантажених артефактів діє як первісний засіб синхронізації виходу проєктів на локальній системі.

Результат виконання maven package для збірки проєкту:



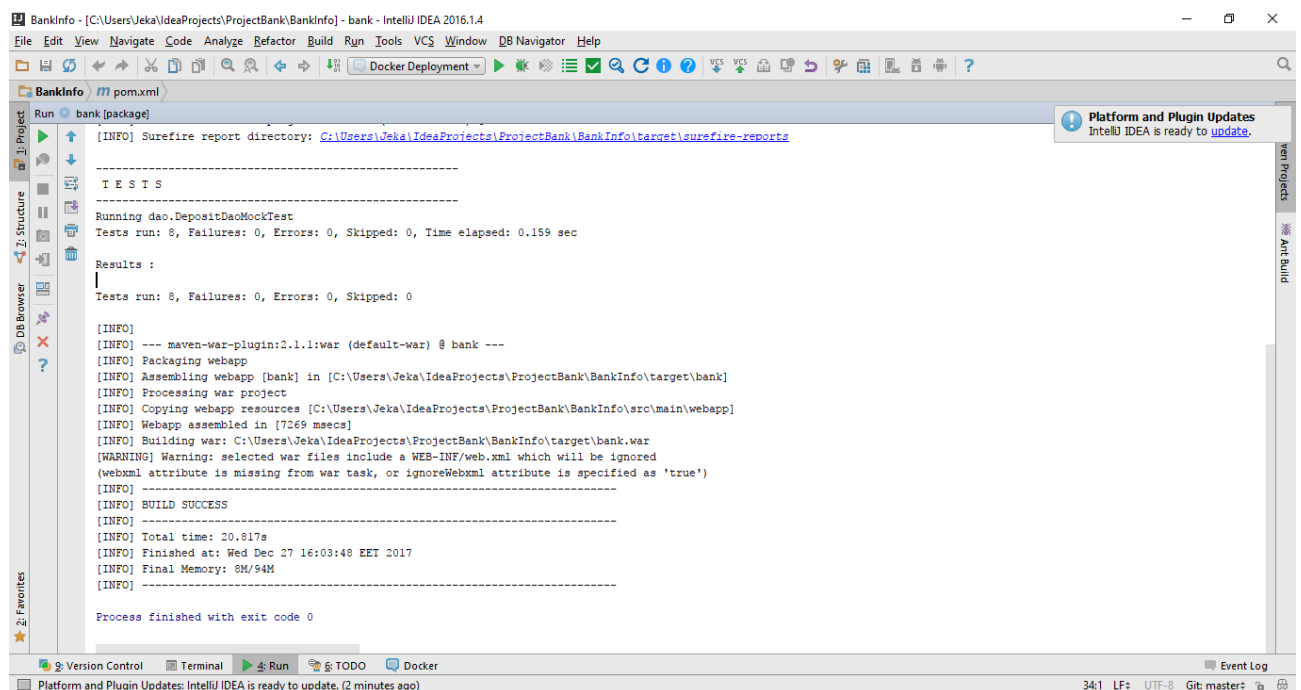
```
BankInfo - [C:\Users\Jeka\IdeaProjects\ProjectBank\BankInfo] - bank - IntelliJ IDEA 2016.1.4
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window DB Navigator Help
BankInfo pom.xml
Run bank [package]
[C:\Program Files\Java\jdk1.8.0_91\bin\java] "-Dmaven.home=C:\Program Files (x86)\JetBrains\IntelliJ IDEA Community Edition 2016.1.3\plugins\maven
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building bank 1.2
[INFO] -----
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ bank ---
[debug] execute contextualize
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ bank ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @ bank ---
[debug] execute contextualize
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\Jeka\IdeaProjects\ProjectBank\BankInfo\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ bank ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.10:test (default-test) @ bank ---
[INFO] Surefire report directory: C:\Users\Jeka\IdeaProjects\ProjectBank\BankInfo\target\surefire-reports

T E S T S

Running dao.DepositDaoMockTest
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.159 sec

Results :

Platform and Plugin Updates: IntelliJ IDEA is ready to update. (2 minutes ago)
16:68 LF UTF-8 Git: master
```



```
BankInfo - [C:\Users\Jeka\IdeaProjects\ProjectBank\BankInfo] - bank - IntelliJ IDEA 2016.1.4
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window DB Navigator Help
BankInfo pom.xml
Run bank [package]
[INFO] Surefire report directory: C:\Users\Jeka\IdeaProjects\ProjectBank\BankInfo\target\surefire-reports

T E S T S

Running dao.DepositDaoMockTest
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.159 sec

Results :

Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[INFO] --- maven-war-plugin:2.1.1:war (default-war) @ bank ---
[INFO] Packaging webapp
[INFO] Assembling webapp [bank] in [C:\Users\Jeka\IdeaProjects\ProjectBank\BankInfo\target\bank]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\Users\Jeka\IdeaProjects\ProjectBank\BankInfo\src\main\webapp]
[INFO] Webapp assembled in [7269 msecs]
[INFO] Building war: C:\Users\Jeka\IdeaProjects\ProjectBank\BankInfo\target\bank.war
[WARNING] Warning: selected war files include a WEB-INF/web.xml which will be ignored
(webxml attribute is missing from war task, or ignoreWebxml attribute is specified as 'true')
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 20.817s
[INFO] Finished at: Wed Dec 27 16:03:48 EET 2017
[INFO] Final Memory: 8M/94M
[INFO] -----

Process finished with exit code 0

Platform and Plugin Updates: IntelliJ IDEA is ready to update. (2 minutes ago)
34:1 LF UTF-8 Git: master
```

3. Сервер безперервної інтеграції. Travis-ci

Як виявилось, термін «continuous integration» досить старий. Він був введений Мартіном Фаулером (Martin Fowler) у 2000-му році і викладений у статті «Continuous Integration» і по-російськи звучить як «безперервна інтеграція». Це частина процесу розробки, в якій розробляється проєкт збирається / тестується в різних середовищах виконання автоматично і безперервно. Задумувалася дана методика для найбільш швидкого

виявлення помилок / протиріч інтеграції проекту, а отже зниження витрат на наступні прості.

Принцип досить простий: на окремій машині працює якась служба, в обов'язки якої входить отримання вихідного коду проекту, його збірка, тестування, логування, а також можливість надати для аналізу дані виконання перерахованих операцій.

Звичайно ж, безкоштовний сир буває тільки в мишоловці і за зручність необхідно платити: виділити окремий сервер і підтримувати його в робочому стані, забезпечити наявність необхідних програмних комплексів, налаштувати середовища виконання, робити резервні копії даних і т.д. Все це вимагає чимало часу і ресурсів. І цілком логічним здається можливість делегувати цю відповідальність на сторонні сервіси. От якраз таким і є travis-ci - «хостинг безперервної інтеграції для open source співтовариства». Настав час подивитися на нього ближче.

Travis-ci підтримує безліч мов програмування серед яких є і ruby (що не дивно, тому що спочатку він розроблявся для ruby-проектів). Почати користуватися сервісом дуже просто. Потрібно всього лише зробити кілька кроків, які детально описані у власному Гайд проекту. Я лише опишу процес в цілому.

Короткий перелік багів, які шукає утиліта:

- Проблеми продуктивності, пов'язані з розміткою інтерфейсу
- Невикористані ресурси
- Невідповідності розмірів масивів (коли масиви визначені у множинних конфігураціях).
- Проблеми доступності та інтернаціоналізації («магічні» рядки, відсутність атрибуту contentDescription і т.д)
- Проблеми з іконками (невідповідності розмірів, порушення DRY)
- Проблеми зручності використання (Наприклад, не зазначений спосіб введення для текстового поля)
- Помилки в маніфесті

4. Експоненціальна витримка

Для вирішення задачі раптового зникнення з'єднання з базою даних було вирішено задачу експоненціальної витримки (для того, щоб не потрібно було виконувати перезапуск сервера). Для цього ми задіяли бібліотеку c3r0.

Формула витримки складеться лише з двох параметрів. Мінімальний час у степені кількості спроб. Отриманий графік можна переглянути нижче

