Міністерство освіти і науки України

Департамент освіти і науки Вінницької облдержадміністрації

Вінницьке територіальне відділення МАН України

Відділення: «Комп’ютерні науки»

Секція: «Мультимедійні системи, навчальні програми та ігрові програми»

**Проєктування мультимедійного додатка**

Роботу виконав:

дійсний член МАН

Мальований Денис Олегович,

учень 10 класу

комунального закладу

“Вінницький технічний ліцей”

Науковий керівник:

Єфременюк Ніна Василівна,

учитель інформатики

комунального закладу

“Вінницький технічний ліцей ”,

учитель-методист

Вінниця – 2020

ЗМІСТ

[**ВСТУП** 3](#_Toc32611365)

[**РОЗДІЛ 1 ВИБІР МОВИ І СЕРЕДОВИЩА ПРОГРАМУВАННЯ** 7](#_Toc32611366)

[**РОЗДІЛ 2 СТВОРЕННЯ МОДЕЛІ** 8](#_Toc32611367)

[**2.1. Структура програми** 8](#_Toc32611368)

[**2.2. Клас Space** 9](#_Toc32611369)

[**2.3. Клас Database** 10](#_Toc32611370)

[**2.4. Клас Interface** 11](#_Toc32611371)

[**2.5. Клас Client** 12](#_Toc32611372)

[**2.6. Клас Server** 13](#_Toc32611373)

[**2.7. XML-файл** 14](#_Toc32611374)

[**2.8. Як усе працює** 15](#_Toc32611375)

[**РОЗДІЛ 3 ОПИС РОБОТИ ПРОГРАМИ** 16](#_Toc32611376)

[**ВИСНОВКИ** 19](#_Toc32611377)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** 20](#_Toc32611378)

**ВСТУП**

Ідеєю науково-дослідної роботи є створення програмного забезпечення для дослідження зоряних систем та взаємодії їх компонентів, що буде корисний як для звичайного користувача, так і науковця в даній галузі.

Дослідження моделей зоряних систем доволі **актуально**, оскільки у наш час дуже стрімкого розвитку зазнає космічна галузь: запуск штучних супутників стає все буденнішим, дешевшим і простішим. Науковці планують колонізувати найближчі планети, досліджують найвіддаленіші закутки всесвіту. Але велика кількість людей не уявляє за якими законами рухаються космічні об’єкти.

**Основною метою** науково-дослідної роботи є створення кросплатформного мультимедійного додатка для демонстрації руху й взаємодії космічних тіл, дослідження орбітальної механіки, а також гри із друзями.

**Предметом дослідження** є зоряні системи та їх компоненти: зорі, планети, супутники й інші космічні тіла.

**Об’єктом дослідження** є фізичні та математичні взаємозв’язки між космічними тілами й складовими зоряних систем.

**Прикладна цінність** науково-дослідної роботи полягає у можливості використання програми для навчання, експериментів і розваг.

**Очікувані результати виконання роботи:**

* дослідити математичну модель взаємодії космічних тіл та її закони;
* покращити вміння роботи з графічною бібліотекою OpenGL на С++ та Processing (Java);
* покращити вміння роботи із системою контролю версій Git на базі веб-платформи gitlab.com;
* покращити вміння застосовувати ООП;
* покращити вміння застосовувати мультипотокові застосунки;
* покращити вміння застосовувати мережеві технології;
* покращити вміння застосовувати алгоритми довгої арифметики ;
* покращити вміння розробки кросплатформних додатків мовою програмування С++;
* розробити прикладну програму для мультимедійного зображення дослідженої моделі й взаємодії із нею.

Велике натхнення на створення проєкту було отримано від гри Kerbal Space Program. На відміну від неї, даний проєкт не призначений для створення власної льотної техніки та виконання місій, має менше різноманітних механік, проте дозволяє досліджувати зоряні системи разом із друзями, у мультиплеєрі.

**Теоретичні відомості для побудови моделі**

Для побудови фізично-математичної моделі проєкту мені знадобились підручник за 9 клас для поглибленого вивчення математики, а саме таких тем:

* вектори;
* рівняння фігур (еліпс);
* рівняння прямої на координатній площині;

і фізики:

* рівнозмінний рух;
* закони Ньютона;
* закон всесвітнього тяжіння;

З наукових джерел візьмемо:

* Кеплерові елементи орбіти;
* Закони Кеплера (наслідки законів Ньютона).

Використаємо три закониНьютона:

* *“Існують такі системи відліку, в яких тіло перебуває у стані спокою або рівномірного прямолінійного руху, коли на нього не діють інші тіла або дія цих тіл скомпенсована”*;
* *“Прискорення, з яким рухається тіло в інерціальних системах відліку, прямопропорційне рівнодійній силі, що діє на тіло, і обернено пропорційне масі тіла. ”*;
* *“Усі тіла в інерціальних системах відліку взаємодіють між собою із силами, що направлені вздовж однієї прямої, рівні за модулем та протилежні за напрямком*” (справедливо для пари тіл);

також **закон всесвітнього тяжіння**: “Усі тіла взаємодіють між собою із силою, що прямопропорційна добутку мас цих тіл і обернено пропорційна квадрату відстані між ними. Ця сила напрямлена вздовж прямої, що сполучає центри мас тіл. ”;

і перший із **трьох законів Кеплера**: *“Всі планети обертаються навколо Сонця еліптичними орбітами, в одному з фокусів яких перебуває Сонце (всі орбіти планет і тіл Сонячної системи мають один спільний фокус, в якому, власне, і розташовано Сонце)”*.

Найближча до Сонця точка орбіти називається **перигелієм (перицентром)**, а найдальша від нього точка — **афелієм (апоцентром)**.

Для задання початкових векторів положення та руху небесних тіл використаємо **Кеплерові елементи орбіти**, які конвертуємо в **декартові вектори** за алгоритмом, що вказаних у посиланнях.

**РОЗДІЛ 1  
ВИБІР МОВИ І СЕРЕДОВИЩА ПРОГРАМУВАННЯ**

Для прототипування архітектури ПЗ я використав мову і середовище програмування Processing, що базується на Java і має вбудовану графічну бібліотеку OpenGL. Вона надає багато можливостей при роботі з графікою, а багата документація й різноманіття функцій робить процес розробки значно легшим.

Сильні сторони Processing:

* легкий і простий;
* має вбудовану графічну бібліотеку OpenGL, проте назви й використання функцій значно простішим;
* кросплатформний;
* швидкий;
* гнучкий.

Java використовує JVM (Java Virtual Machine) – віртуальну машину Java. Завдячуючи цьому з’являються перспективи використання програми на різних ОС. Код транслюється в байт-код віртуальної машини, завдяки чому він не залежить від виконавчої платформи, а завдяки численним оптимізаціям, програми на Java працюють з чудовою швидкодією для такого рівня абстракції.

Для кінцевої реалізації архітектури ПЗ використана мова програмування C++ разом із графічною бібліотекою OpenGL. Такий варіант використовує в рази менше пам’яті, має в рази більшу швидкодію, а також графічні покращення.

**РОЗДІЛ 2  
СТВОРЕННЯ МОДЕЛІ**

**2.1. Структура програми**

Архітектура ПЗ складається з таких класів: **Interface**, **Space**, **Body**, **Server**, **Client**. Графічно її можна представити так (Рис. 2.1).

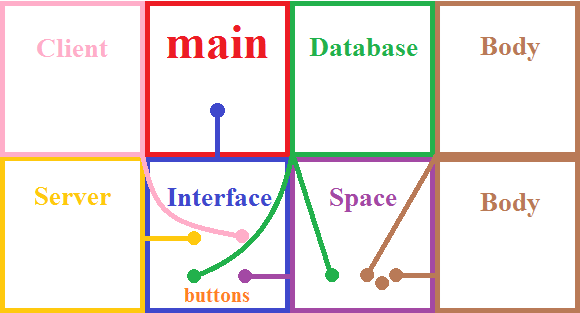


Рис.2.1 Архітектура проєкту

**2.2. Клас Space**

Клас **Space** створений для керування симуляцією. Містить підклас **Body** – модель небесного тіла. Найцікавішим методом класу **Space** є **generate()**, який викликає **generateSpace()** – рекурсивна функція генерування об’єкта класу **Body** з файлу XML. Останній метод, у свою чергу, викликає **convertKeplerianToCartesian()**, що конвертує Кеплерові елементи орбіти в декартові вектори.

**2.3. Клас Database**

Клас **Database** створений для роботи зі статичними файлами: текстури, аудіо та XML. Цікаво, що для збільшення швидкодії об’єкт класу може запам’ятовувати список файлів.

**2.4. Клас Interface**

Клас **Interface** створений для керування симуляцією, взаємодією з користувачем. Має підклас **Button**, який реалізує кнопку.

**2.5. Клас Client**

Клас **Client** створений для комунікації у комп’ютерних мережах, тобто багатокористувацької гри.

**2.6. Клас Server**

Клас **Server** створений для керування багатокористувацькою грою. Він наслідує **Client**.

**2.7. XML-файл**

Початкові дані про світ зберігаються у відповідному XML-файлі ієрархічно. Тобто між тегами одного небесного тіла розташовується інформація про його супутники. Приклад файлу Сонячної системи (Рис.2.6.1, Рис.2.6.2, Рис.2.6.3).



Рис.2.6.1



Рис.2.6.2



Рис.2.6.3

**2.8. Як усе працює**

Головна функція програми створює об’єкт класу **Interface**, який реалізує головне меню програми: гра, редактор та титри. За вибраним користувачем файлом він створює об’єкт класу **Space**, що генерує ігровий світ, створюючи об’єкти класу **Body**. Цей же клас ініціалізує сервер для гри у мультиплеєрі. Після завершення гри чи редагування світу можна повернутися в головне меню.

**РОЗДІЛ 3  
ОПИС РОБОТИ ПРОГРАМИ**

Після запуску гра зустрічає Вас у головному меню з анімованим фоном (Рис.3.1). Ви можете перейти до екрана вибору світу (Рис.3.2) для гри або редагування, а також титрів (Рис.3.3). Ви можете обрати режим поодинокої або багатокористувацької гри.



Рис.3.1 Головне меню

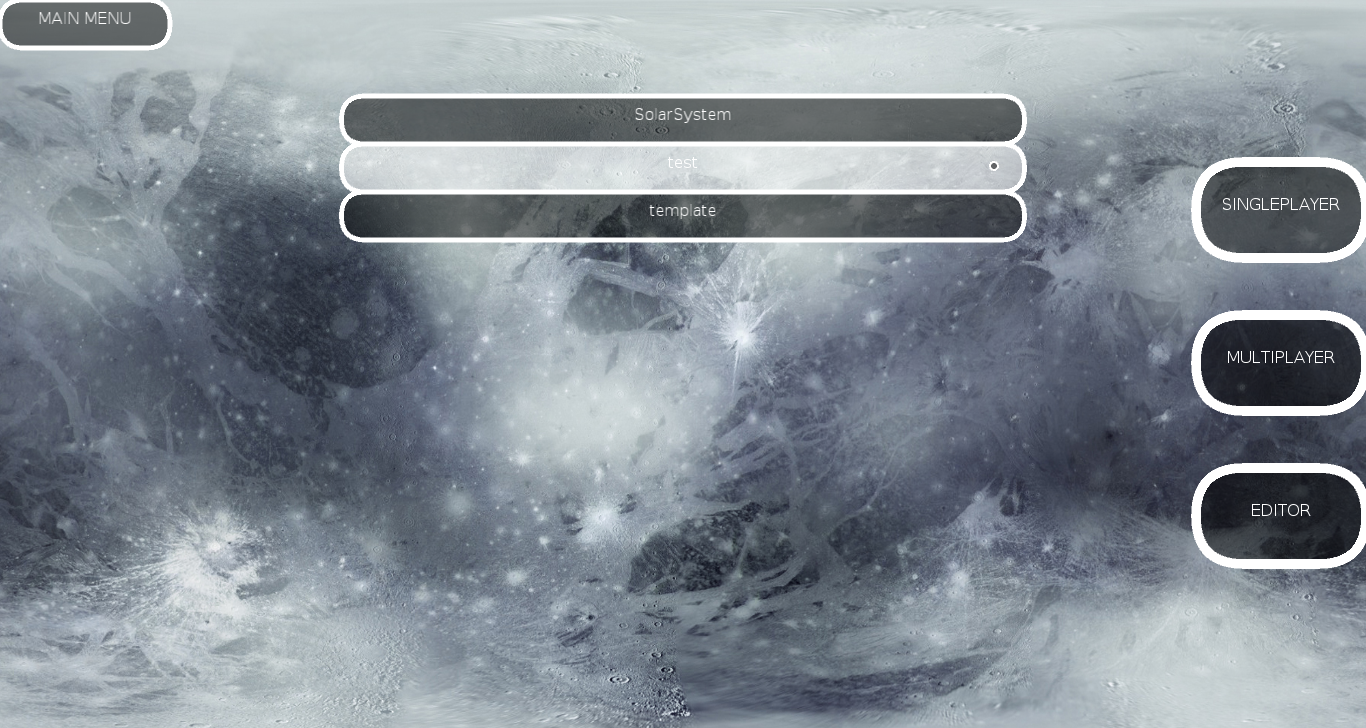


Рис.3.2 Екран вибору світу

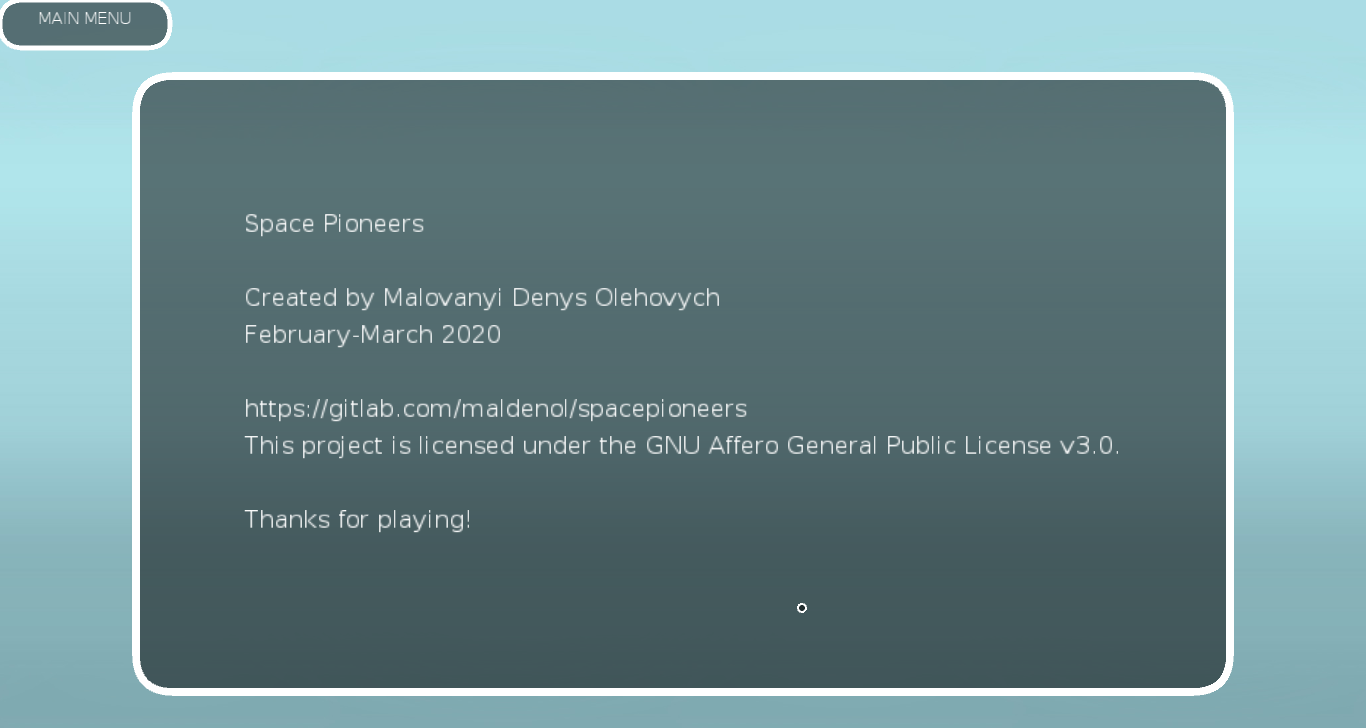


Рис.3.3 Титри

Керування положенням камери спостерігача відбувається клавішами ‘w’, ‘a’, ‘s’, ‘d’, ‘q’, ‘e’ для руху вгору, вліво, вниз, вправо, назад і вперед відповідно. Клавішею ‘c’ можна перемикатись між різними видами: вільна камера, об’єктив телескопа, від третьої особи. Кнопка ‘m’ перемикає зображення текстур планет, ‘i’ – інтерфейсу, а ‘o’ – орбіт планет. Кнопка ‘p’ зупиняє і продовжує хід симуляції. У правому верхньому кутку показано зверху донизу відповідно FPS (кількість кадрів за секунду).

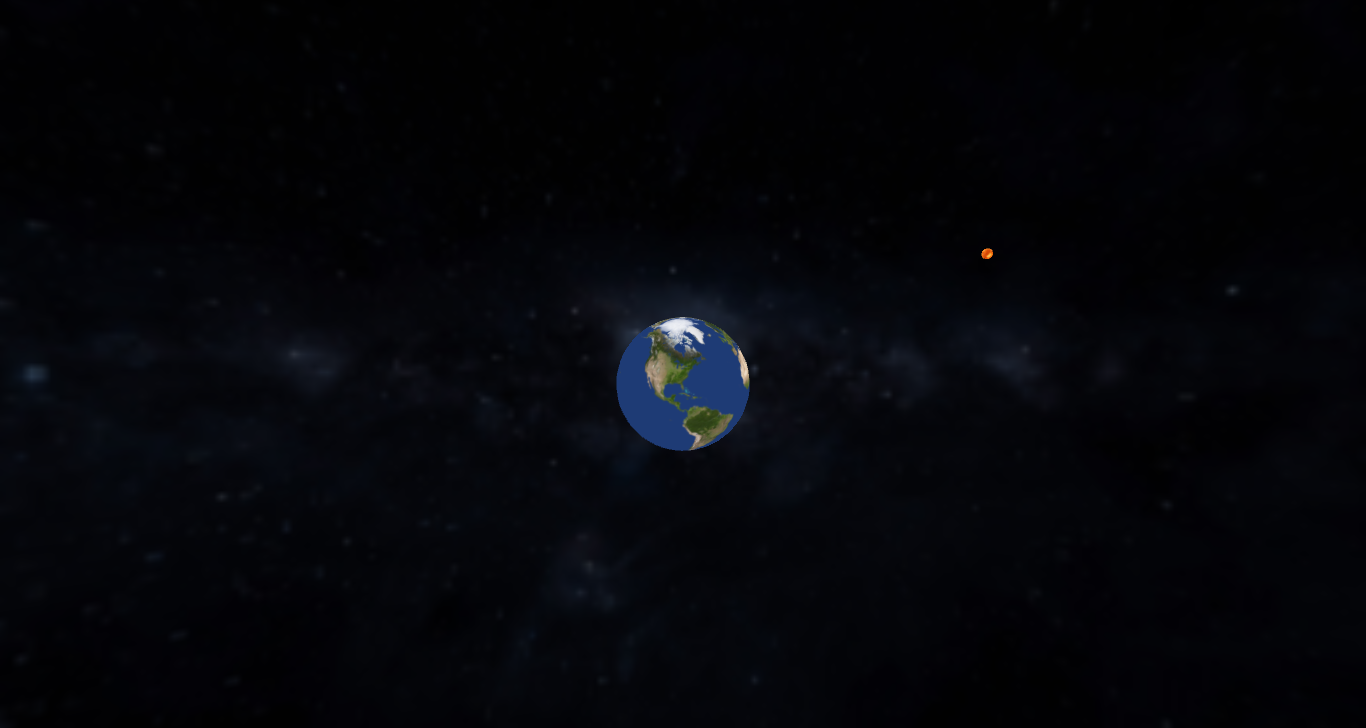


Рис.3.4 Тестовий світ

**ВИСНОВКИ**

Під час виконання проєкту я набув навичок дослідження фізичних та математичних моделей Всесвіту, розробки та налагодження програм, здобув досвід роботи з СКВ Git, мережевими технологіями, довгою арифметикою, мультимедійними бібліотеками, мовами програмування різного рівня абстракції.

Була ретельно досліджена та програмно реалізована модель зоряних систем. Створено інтерфейс для взаємодії з симуляцією, з можливістю завантаження, генерації й редагування власної зоряної системи. Реалізовано можливість керування власним космічним телескопом у режимі мультиплеєру.

Використовуючи розроблену програму люди навчаються, експериментують, набувають знань у сфері астрономії й фізики. Крім того, ПЗ можна використовувати для науково точної симуляції зоряних систем.

Результат виконаної роботи можна завантажити за посиланням <https://gitlab.com/maldenol/spacepioneers/>. Початковий код програми ліцензований під GNU AGPLv3.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Wolff D. OpenGL 4 Shading Language Cookbook: Build high-quality, real-time 3D graphics with OpenGL 4.6, GLSL 4.6 and C++17 / David Wolff., 2018. – 472 с. – (3).; Packt Publishing; ISBN 978-1-78-934225-3.
2. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3 / D. Shreiner, G. Sellers, J. Kessenich, B. Licea-Kane., 2013. – 984 с. – (8).; Addison-Wesley Professional; ISBN 978-0-32-177303-6.
3. Kernighan B. W. C Programming Language / B. W. Kernighan, D. M. Ritchie., 1988. – 272 с. – (2).; Prentice Hall; ISBN 978-0-13-110362-7.
4. Stroustrup B. The C++ Programming Language / Bjarne Stroustrup., 2013. – 1376 с. – (4).; Addison-Wesley Professional; ISBN 978-0-32-156384-2.
5. Stroustrup B. Programming: Principles and Practice Using C++ / Bjarne Stroustrup., 2014. – 1312 с. – (2).; Addison-Wesley Professional; ISBN 978-0-32-199278-9.
6. Schildt H. C++: The Complete Reference / Herbert Schildt., 2002. – 1056 с. – (4). – (Osborne Complete Reference Series).; McGraw-Hill Education; ISBN 978-0-07-222680-5.
7. Meyers S. Effective C++: 55 Specific Ways to Improve Your Programs and Designs / Scott Meyers., 2005. – 320 с. – (3).; Addison-Wesley Professional; ISBN 978-0-32-133487-9.
8. Meyers S. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14 / Scott Meyers., 2014. – 334 с.; O'Reilly Media; ISBN 978-1-49-190399-5.
9. Alexandrescu A. Modern C++ Design: Generic Programming and Design Patterns Applied / Andrei Alexandrescu., 2001. – 360 с.; Addison-Wesley Professional; ISBN 978-0-20-170431-0.
10. Царьков М. небесная механика [Електронний ресурс] / Михайло Царьков – Режим доступу до ресурсу: <https://www.youtube.com/watch?v=fBC3cCsCEzA>.
11. Alpha Centauri. Всё об орбитальной механике | Как запускают спутники [Електронний ресурс] / Alpha Centauri – Режим доступу до ресурсу: <https://www.youtube.com/watch?v=YvbB4S5NiX8>.
12. Schwarz R. Keplerian Orbit Elements → Cartesian State Vectors [Електронний ресурс] / René Schwarz // 1. – 2017. – Режим доступу до ресурсу: <https://downloads.rene-schwarz.com/download/M001-Keplerian_Orbit_Elements_to_Cartesian_State_Vectors.pdf>.
13. Shiffman D. The Nature of Code: Simulating Natural Systems with Processing / Daniel Shiffman.
14. Shiffman D. Learning Processing, Second Edition: A Beginner's Guide to Programming Images, Animation, and Interaction / Daniel Shiffman.
15. Reas C. Processing: A Programming Handbook for Visual Designers, Second Edition / C. Reas, B. Fry.
16. Reas C. Make: Getting Started with Processing, Second Edition / C. Reas, B. Fry.
17. Schildt H. Java - The Complete Reference / Herbert Schildt., 2014. –Reas C. Reference. Processing was designed to be a flexible software sketchbook.
18. [Електронний ресурс] / C. Reas, B. Fry, D. Shiffman. – 2019. – Режим доступу до ресурсу: <https://processing.org/reference/>.
19. Клімов О. Освой программирование играючи [Електронний ресурс] / Олександр Клімов. – 2019. – Режим доступу до ресурсу: <http://developer.alexanderklimov.ru/>.
20. ORACLE Java Documentation [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://docs.oracle.com/javase/tutorial/.