

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук,  
управління та адміністрування  
Кафедра Інформаційних технологій

**БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему: Розробка застосування з ігровими елементами для навчання  
друку на клавіатурі

Виконав студент 4 курсу групи К-41  
Спеціальність 122 Комп'ютерні науки  
Ряшенцев Вадим Сергійович

Керівник ст.викл.  
Рольщиков Вадим Борисович

Консультант к.геогр.н., доцент  
Кузніченко Світлана Дмитрівна

Рецензент к.геогр.н., доцент  
Лужбін Анатолій Михайлович

## ЗМІСТ

Перелік скорочень, умовних позначень і термінів .....	6
Вступ.....	7
1 Аналітична частина.....	9
Моделювання процесу розробки додатку .....	9
Аналіз предметної області .....	11
Аналіз видів та структур клавіатурф .....	12
Аналіз мембранних клавіатур.....	12
Аналіз механічних клавіатур .....	13
Аналіз напівмеханічних клавіатур .....	14
Аналіз ергономічних клавіатур .....	15
Аналіз ринку подібних проектів .....	17
Розгляд клавіатурного тренажера Соло на клавіатурі.....	17
Розгляд клавіатурного тренажера VerseQ.....	20
Розгляд клавіатурного тренажера Stamina.....	22
Аналіз інструментів розробки .....	24
Вибір платформи розробки.....	25
Опис .NET.....	25
Опис Java .....	26
Вибір версії ПЗ.....	28
Вибір СУБД.....	29
Опис MySQL .....	29
Опис SQLite .....	30
Вибір IDE .....	32
Розгляд Eclipse .....	32
Розгляд IntelliJ Idea .....	33
2 Проектна частина .....	35
Постановка завдань .....	35
UML діаграма проекту .....	35

	5
Опис основних класів .....	35
Спільна частина .....	37
Навчальна частина .....	37
2.3.2 Ігрова частина.....	40
3 Опис окремих програмних елементів розробки.....	44
Опис ігрових елементів .....	44
Опис руху герою .....	44
Опис руху злодія .....	45
Опис клавiш.....	46
Опис навчальних елементів .....	47
Текстові:.....	47
Аудіювання.....	48
Висновки .....	49
Перелік джерел посилання .....	50
Додаток А. Код програми.....	52

**ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ**

БД – база даних

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

ПП – програмний продукт

СУБД – система управління базами даних

JVM – Java Virtual Machine, віртуальна машина Java

IDE – Integrated Development Environment, інтегроване середовище розробки.

## ВСТУП

Оскільки в сучасне життя усе глибше і глибше проникають комп'ютерні технології, то хочемо ми того чи ні а приходиться освоювати нові, досі невідомі раніше практичні навички.

Якщо ази комп'ютерної грамотності вже є у вашому особистому багажі, то настає час для розвитку "швидкості" і більш ефективної роботи з таким воістину багатогранним інструментом як клавіатура. І хоча потужність комп'ютера росте з кожним роком, швидкість роботи з ним обмежується лише людським фактором.

На сьогоднішній день ще не придумано більш швидкого способу введення даних у комп'ютер, чим старий-добрий клавіатурний метод. От і приходиться сучасній людині освоювати заодно стародавню професію "друкарки", як і 100 років тому, із дня в день, удосконалюючи навички "сліпого десятипальцевого" набору текстів на клавіатурі.

Звичайно, з роками розташування клавіш "запам'ятовується" пальцями автоматично, і швидкість набору текстів у користувачів ПК зі стажем стає досить високою. Але, як показує практика, їх "сліпий метод" виявляється менш ефективним, чим "класичний". Руки роблять зайві рухи, неправильно лежать на клавіатурі, поступово накопичуючи тим самим зайву утому і викликаючи біль у суглобах.

Щоб початківець-користувач (чи програміст) додав у своє резюме серйозну фразу "сліпий десятипальцевий набір текстів на клавіатурі", і взагалі, істотно прискорив роботу на ПК (звільняючи час для інших занять), була створена достатня кількість спеціалізованих програм – клавіатурних тренажерів.

Метою даної дипломної роботи є моделювання та розробка системи, яка зможе оцінювати рівень підготовки користувача та допоможе йому обрати правильний рівень навчання. Додаток повинен пропонувати усі види нав-

чання, починаючи з найлегшого, та закінчуючи найскладнішим. Коли користувач завершує одну з цих навчальних програм, система повинна аналізувати, та виводити на екран що потрібно робити далі користувачу.

Представлена пояснювальна записка містить 51 сторінку, 27 рисунків, 22 посилання та 20 сторінок додатка.

## 1 АНАЛІТИЧНА ЧАСТИНА

### Моделювання процесу розробки додатку

Моделювання процесу розробки додатку зроблено за допомогою графічного редактору, схожого на IDEF0[1]<sup>1)</sup>.

Проект «Розробка клавіатурного тренажера з елементами гри» у цілому повинен задовольняти такі потреби: навчати користувачів використовувати клавіатуру так, щоб руки не почували втому, суглоби не боліли, підвищити швидкість друку та зменшити кількість помилок при друкуванні великих текстів.

На нульовому рівні (рис. 1) можна побачити на вході (зліва від блоку) – описана поставлена задача, яку потрібно виконати упродовж проекту. Для виконання потрібно мати програмне забезпечення – ці стрілки розташовуються знизу вгору. Стрілки розташовані зверху – означають інформацію, яка потребується для реалізації проекту. Стрілка розташована праворуч від блоку – готовий клавіатурний тренажер.

На першому рівні (рис. 2) розташовані роботи, а саме модулювання предметної області.

Дану проблему можна розділити на декілька структурних одиниць. У даному випадку було прийнято рішення розділити на дві структурні одиниці «розробка ігрової частини» та «розробка навчальної частини», це можна почитати на рис. 3.

В результаті успішного виконання цих структурних одиниць можна отримати застосування, яке буде допомагати користувачу навчатися сліпому друку, також можна відволікатися від навчання та протестувати гру,

На рис 1 проілюстровано усі необхідні ресурси, вихідні дані та необхідний результат даного проекту.

---

<sup>1)</sup> [1] Методология IDEF0. URL: <https://itteach.ru/bpwin/metodologiya-idef0> (дата звернення 15.02.2020).

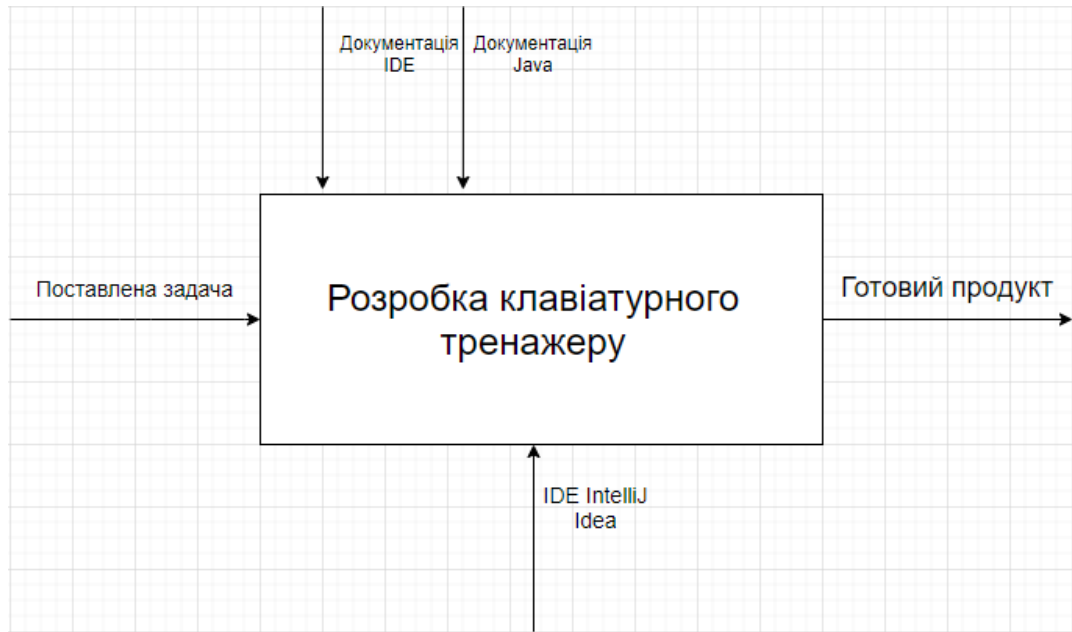


Рисунок 1 – IDEF0 «Розробка клавiатурного тренажеру з елементами гри»,  
нульовий рівень

На рис. 2 проiлюстровано моделювання предметної області.

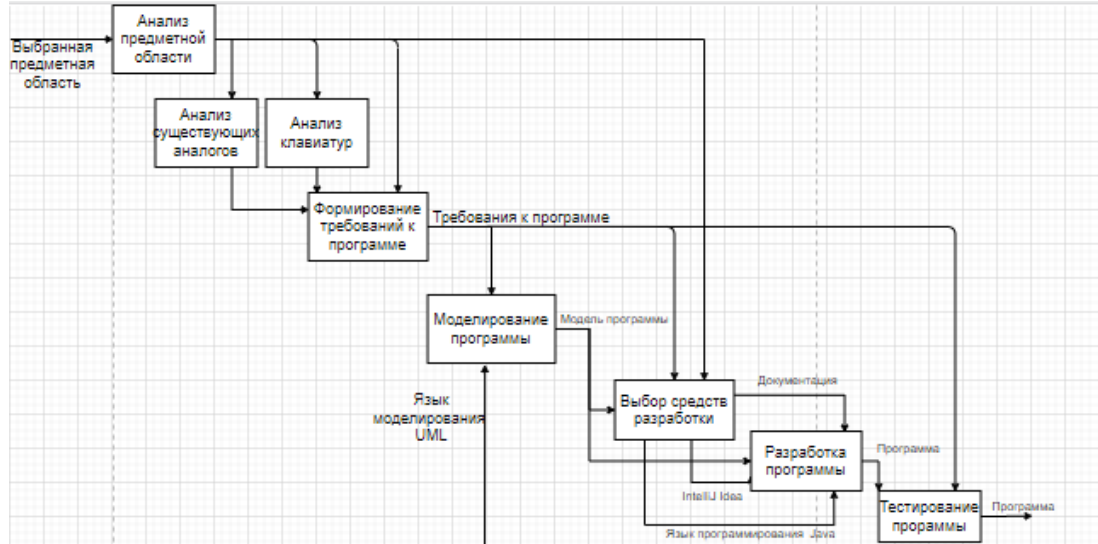


Рисунок 2 – IDEF0 «Розробка клавiатурного тренажеру з елементами гри»,  
перший рівень

На рис. 3 проiлюстровано три роботи, які потрібно виконати для успішної реалізації проекту.



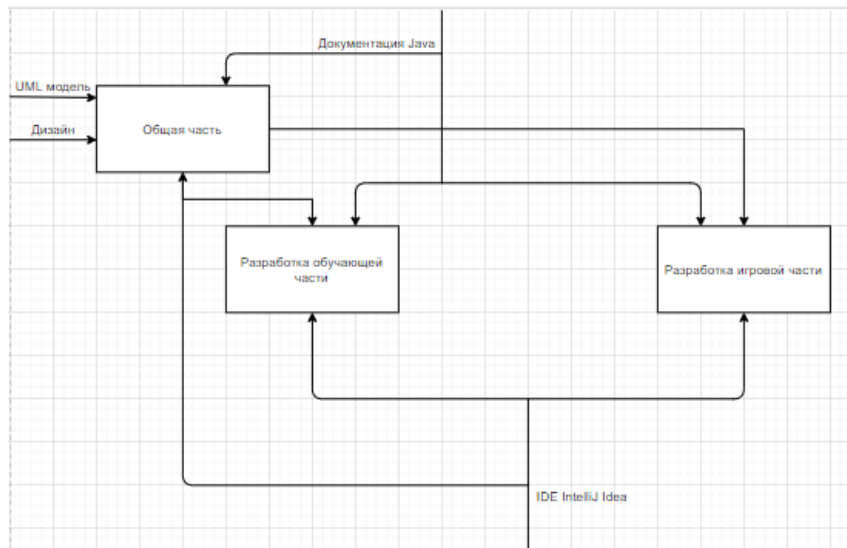


Рисунок 3 – IDEF0 «Розробка клавіатурного тренажера з елементами гри», другий рівень

### Аналіз предметної області

У даному розділі будуть детально розглянути проблеми та завдання, які необхідні виконати у даній частині проекту.

Програмний продукт(сокр. ПП) включатиме в себе місце збереження всіх необхідних даних, а також програми настільного застосування(desktop application), яке відображатиме прогрес попередніх занять при наступному запуску програми.

Під час розробки ПП повинні виконуватися наступні вимоги:

- ПП повинно бути створеним по всім вимогам навчання;
- ПП повинен мати графічний інтерфейс користувача;
- ПП повинен мати БД у якій буде зберігатися вся необхідна інформація для комфортного користування ПП.
- ПП повинно збирати статистику для поліпшення навичок користувача.

Актуальність цієї роботи полягає в тому, що у всіх, хто багато сидить за комп'ютером та постійно друкує, втомлюються руки та болять суглоби. Клавіатурні тренажери повинні вирішити це питання.

Тих, хто вчиться сліпому друку можна поділити на 4 категорії:

- без навичок;
- початківців;
- користувач середнього рівня;
- користувач високого рівня.

Ділення користувачів на категорії потрібно для того, щоб підібрати правильну програму навчання.

Дане програмне забезпечення(сокр. ПЗ) розроблено для людей без навичок та початківців.

## **Аналіз видів та структур клавіатурф**

Сьогодні зустріти величезне розмаїття клавіатур. Клавіатури бувають мембранні[2]<sup>1)</sup>, напівмеханічні[3]<sup>2)</sup>, механічні[4]<sup>3)</sup>, та ергономічні[5]<sup>4)</sup>.

### **Аналіз мембранних клавіатур**

Дана категорія пристроїв користується дуже великим попитом, так як є найоптимальнішими в ціновому співвідношенні, а також для більшості за якістю роботи ні чим не буде відрізняються від інших видів. Йдеться, звичайно, не про тих хто розбирається і знає, що йому потрібна саме механічний тип.

---

<sup>1)</sup> [2] Мембранная клавиатура – Википедия. URL: [https://ru.wikipedia.org/wiki/Мембранная\\_клавиатура](https://ru.wikipedia.org/wiki/Мембранная_клавиатура) (дата звернення 16.01.2020).

<sup>2)</sup> [3] Разновидности и виды клавиатур. URL: <https://procompsoft.ru/stati/vidy-klaviatur.html> (дата звернення 16.01.2020).

<sup>3)</sup> [4] Механическая клавиатура – Википедия. URL: [https://ru.wikipedia.org/wiki/Механическая\\_клавиатура](https://ru.wikipedia.org/wiki/Механическая_клавиатура) (дата звернення 16.01.2020).

<sup>4)</sup> [5] Ергономічна клавіатура. URL: [http://womanwiki.ru/w/Эргономичная\\_клавиатура](http://womanwiki.ru/w/Эргономичная_клавиатура) (дата звернення 16.01.2020).

Цей вид, як правило, виготовляється з таких матеріалів, як: пластик і гума. Сам корпус виконаний з пластмаси, кнопки вставлені в свої місця за коштом засувки під цим простацькі механізмом розташовується плівка, як правило полімерна.

Зовнішній вигляд цієї клавіатури представлений на рис. 1.1.



Рисунок 1.1 – Вигляд мембранної клавіатури

На підставі всієї конструкції можна зробити найголовніший висновок – цей механізм добре захищений від різного роду пошкоджень і попадання сторонніх часток, речовин, які б могли порушити весь процес працездатності. Відмінною особливістю даної моделі є той факт, що натискаючи на кнопку, користувачу доведеться робити це до кінця, тобто до повного зіткнення контактів, на це постійно потрібно хоч і не великі, проте зайві зусилля.

Є і ще невеликий мінус – працюючи з даними клавіатурами, особливо з друкованим документом, слід звертати увагу на екран, так як в роботі такої моделі можливо і заїдання дії кнопки, так само вона менш надійна ніж наступні види клавіатур.

### **Аналіз механічних клавіатур**

Суть роботи даних моделей полягає в тому, що увесь принцип заснований на дії пружини. Простіше кажучи, при натисканні на кнопку, основа пружинної конструкції складається і відбувається натискання клавіши.

Приклад таких клавіатур зображено на рис. 1.2.



Рисунок 1.2 – Наглядний приклад механічних клавіатур

Потім сигнал клавіши передається на монітор, кнопка опускається вниз, а пружина вертається у своє початкове положення. Тобто у нас справжні контакти та пружний механізм, причому для того, щоб спрацював клік зовсім необов'язково натискати клавішу до упору, що порядком може економити сили і нерви, на відміну від повсюдного мембранного засилля де доводиться докласти зусилля постійно інакше ми не чого не виведемо на екран. Технологія досить проста і зрозуміла.

Але, є мінус – це те, що сам механізм не захищений, тобто в комплектації не передбачено спеціального корпусу, який міг би забезпечити всю технологію роботи. Тут мова конкретно про сам механізм, він більш уразливий в порівнянні з іншими видами до вологи й бруду.

Ці клавіатури відносяться до вищого класу, та настійно рекомендуються професійним друкарям[6]<sup>1)</sup>.

### **Аналіз напівмеханічних клавіатур**

За своєю суттю, це різновид механічних клавіатур, однак істотна відмінність його – даний різновид моделей містить у собі не пружини, а пружні гу-

---

<sup>1)</sup> [6] Механическая клавиатура – Википедия. URL: [https://ru.wikipedia.org/wiki/ Механическая\\_клавиатура](https://ru.wikipedia.org/wiki/Механическая_клавиатура) (дата звернення 16.01.2020).

мові елементи або їх подібність, які відповідають за технологію і метод повернення кнопки в початкове, вихідне положення. Плюсом даної моделі є те, що клавіші цих клавіатур довго не зтираються.

На рис. 1.3 зображено наглядний приклад цієї клавіатури. Також можна побачити, що у неї є підставка для рук[7]<sup>1)</sup>.



Рисунок 1.3 – Зовнішній вид напівмеханічної клавіатури

### **Аналіз ергономічних клавіатур**

Ергономічна клавіатура[8]<sup>2)</sup> – це така клавіатура, що має вид дуги, саме поле розбите на частини, своя під кожну руку, і наділена "підставкою на відпочинок рук". Вона, звісно, вродлива, але, ця ергономіка з її підставкою займає значну частину місця на столі, тоді як у інших людей, які не володіють способом сліпого друку, доводиться постійно бігати очима з однієї половини поля в іншу, постійно змінюючи фокусну відстань кришталіка, що стомлює очі.

---

<sup>1)</sup> [7] Разновидности и виды клавиатур. URL: <https://procompsoft.ru/stati/vidy-klaviatur.html> (дата звернення 16.01.2020).

<sup>2)</sup> [8] Ергономічна клавіатура. URL: [http://womanwiki.ru/w/Эргономичная\\_клавиатура](http://womanwiki.ru/w/Эргономичная_клавиатура) (дата звернення 16.01.2020).

При роботі на комп'ютері руки і пальці втомлюються, що призводить до поступового зниження швидкості набору тексту і збільшення кількості помилок. Суттєво зменшити стомлюваність допоможе правильно підібрана ергономічна клавіатура. Дослідження показали, що її використання дозволяє зменшити втому пальців приблизно на 20%, швидкість набору підвищується до 15%, а число помилок зменшується на третину.

На рис. 1.4 зображено приклад цієї клавіатури.



Рисунок 1.4 – Вигляд ергономічної клавіатури

Ергономічна клавіатура відрізняється від звичайної своєю вигнутою формою і нестандартним алфавітним блоком. Особливе розташування клавіш, їх форма, наявність підставки під зап'ястя – всі ці удосконалення призначені зробити введення інформації через цей пристрій максимально зручним. Досить часто ергономічна клавіатура ділиться на дві половинки, під кожен з рук. Іноді можна зустріти навіть такі моделі, де ці два блоки можуть розсуватися, або бути повністю роздільними. Така властивість дуже зручно для користувачів, котрі володіють «сліпим» методом набору тексту. Однак не химерна форма робить її зручною в роботі, а можливість налаштувати цей пристрій під користувача, задавши необхідний кут нахилу. Це дозволяє мінімізувати відстань,

необхідну для переміщення кисті та пальців при натисканні тієї чи іншої клавіші.

Розробка програмного забезпечення передбачено для найпростішої, самої дешевої клавіатури.

### **Аналіз ринку подібних проектів**

Був проведений моніторинг існуючого програмного забезпечення (ПЗ), призначеного для допомоги усім, хто використовує комп'ютери для друку, в їх роботі. В результаті, було виявлено, що є багато програм для навчання сліпого друку. Прикладами таких програм можуть слугувати:

- «Соло на клавіатурі» [9]<sup>1)</sup>;
- «VerseQ» [10]<sup>2)</sup>.
- «Stamina» [11]<sup>3)</sup>;

### **Розгляд клавіатурного тренажеру Соло на клавіатурі**

«Соло на клавіатурі» вважається як система клавіатурного тренажера так і система психологічного тренінгу. Насправді тут немає нічого дивного, адже її автор – психолог Володимир Шахіджян [12]<sup>4)</sup>.

Кожен урок в системі навчання «Соло на клавіатурі» супроводжується різними текстами – порадами, як вчитися за цією програмою, наводяться також витяги з листів користувачів, з текстами і анекдотами й, головне словами, на думку автора, повинні змусити кожную людину повірити в себе, у можливість повністю пройти курс навчання і навчитися друкувати «наосліп».

---

<sup>1)</sup> [9] Курс Соло на клавіатурі. URL: <https://ergosolo.ru/> (дата звернення 10.02.2020).

<sup>2)</sup> [10] VerseQ – клавіатурний тренажер. URL: <https://www.verseq.ru/> (дата звернення 10.02.2020).

<sup>3)</sup> [11] Stamina клавіатурний тренажер для бабусь, дідусів а також їх родичів. URL: <https://stamina.ru/> (дата звернення 10.02.2020).

<sup>4)</sup> [12] Шахиджанян, Володимир Володимирович – Wikipedia. URL: [https://ru.wikipedia.org/wiki/Шахиджанян,\\_Владимир\\_Владимирович](https://ru.wikipedia.org/wiki/Шахиджанян,_Владимир_Владимирович) (дата звернення 10.02.2020).

Ці тексти можна не читати, а виконувати тільки вправи, але автор стверджує, що в цьому випадку ефективність навчання помітно знизиться, так як супроводжуючі тексти представляють таку ж важливу частину занять, як і вправи. Важливо ще одне, Володимир Шахіджаян радить читати абсолютно всі тексти уважно і вдумливо, не поспішаючи. До речі, поспішати не рекомендується навіть при друку тексту – при здійсненні помилки програма скаже фразу «п'ять секунд терпіння».

Перед початком навчання в «Соло на клавіатурі» варто пройти вступний іспит. На самому початку користувачеві пропонують набрати текст так, як він вміє. Програма відразу зафіксує швидкість набору, наявність помилок, а також інші особливості набору. Після завершення навчання можна буде набрати той же текст і порівняти результати. До речі, цей текст не зовсім простий – це перший ваш тренінг, користувачеві доведеться пройти в програмі, надрукувавши «жити треба НЕ поспішаючи, ми всі хочемо жити цікаво» та так далі у тому ж руслі.

Навчання по системі «Соло на клавіатурі» побудовано таким чином, щоб учень не відчував, що він займається сам, наодинці з безликою програмою. Автор «Соло» ніби розмовляє з кожним учнем, намагаючись показати, що вчитель поруч і готовий допомогти у вирішенні проблем, але тільки не тих, які не пов'язані з навчанням десятипальцевому методу. Вікно, на якому виконуються вправи, містить в собі не тільки віртуальну клавіатуру, вікна статистики і повзунок, за яким можна визначити, як далеко користувач просунувся в навчанні, а й крилатий вислів будь-якої відомої людини, який кожен раз вибирається в абсолютно випадковому порядку.

На рис. 1.5 представлений стартовий фрейм, на якому зображене вікно реєстрації у програмі, де користувач може побачити, що у програмі можна обирати різні мови, такі як: англійська, російська та транслітерація.



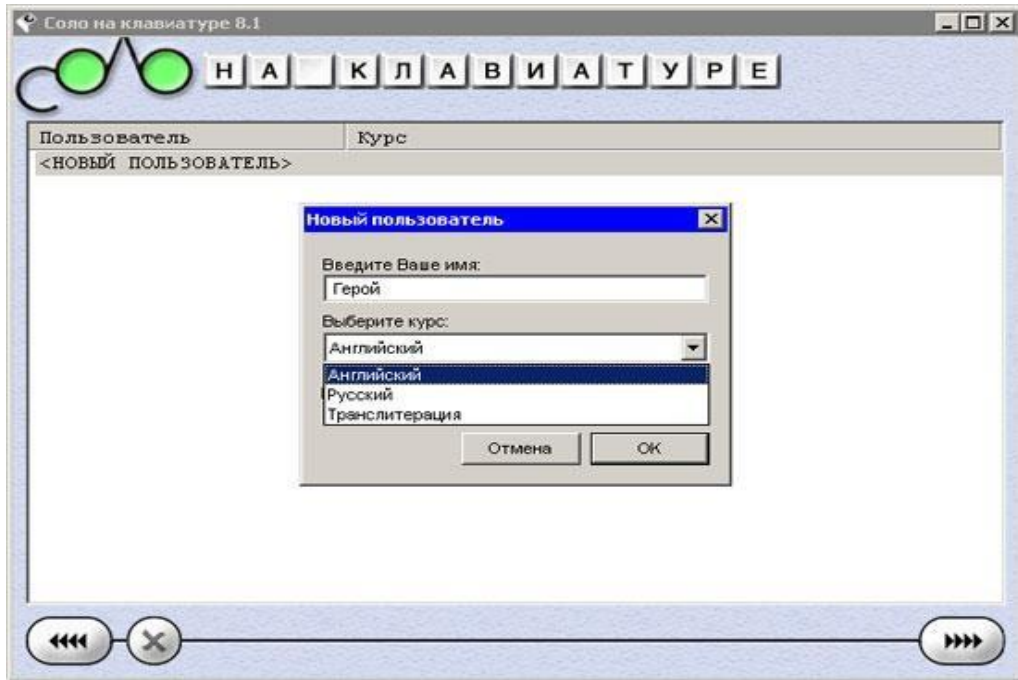


Рисунок 1.5 – Вікно реєстрації Солю на клавіатурі

На рис. 1.6 представлений приклад крилатої фрази, та вид заняття для початківця.



Рисунок 1.6 – Вигляд навчання та крилатих фраз

## Розгляд клавіатурного тренажеру VerseQ

Програма VerseQ[13]<sup>1)</sup> – це поліпшена версія клавіатурного тренажера, раніше відомого як KeyTO. Найперше, чим ця версія «чіпляє» користувача, – це обіцянка автора Володимира Кіма навчити друкувати наосліп вже через годину після початку роботи з програмою. Якщо вже знайомі з традиційним способом навчання десятипальцевого набору, то таке твердження напевно викличе недовіру, адже як правило користувача починають навчати з однієї або двох букв, тому впродовж години він може освоїти максимально 5-6 букв.

Методика Володимира Кіма зовсім інша. У цій системі користувач починає не з двох букв, а відразу з усіх основних (за винятком початкових і спеціальних символів). Автор радить Не поспішай і натискати клавішу лише в тому випадку, якщо учень повністю впевнений, що під його пальцем знаходиться якраз і клавіша, на яку необхідно натиснути. При цьому категорично забороняється дивитися на клавіатуру, навіть рекомендується встановити спеціальну заслінку, щоб коли опускалися очі клавіші Чи не буде видно. Замість цього пропонується віртуальна клавіатура, яка розташовується під рядком тексту, Який необхідно роздрукувати.

У цій програмі не доведеться набирати смислові тексти, хоча іноді можуть попадатися справжні слова. Навчання побудовано на заучуванні так званих акордів, тобто буквосполучень, які зустрічаються в мові найчастіше. Користувачу доведеться друкувати щось по типу «сказыщутились» і «раблялась».

На рис. 1.7 зображене більш яскраве стартове вікно та більш розширений вибір мов.

---

<sup>1)</sup> [13] VerseQ – клавіатурний тренажер. URL: <https://www.verseq.ru/> (дата звернення 10.02.2020).



Рисунок 1.7 – Стартовое окно Verseq

На рис. 1.8 зображене які фрази використовуються у програмі та поточну статистику, а саме: швидкість друку, ритмічність та кількість помилок.

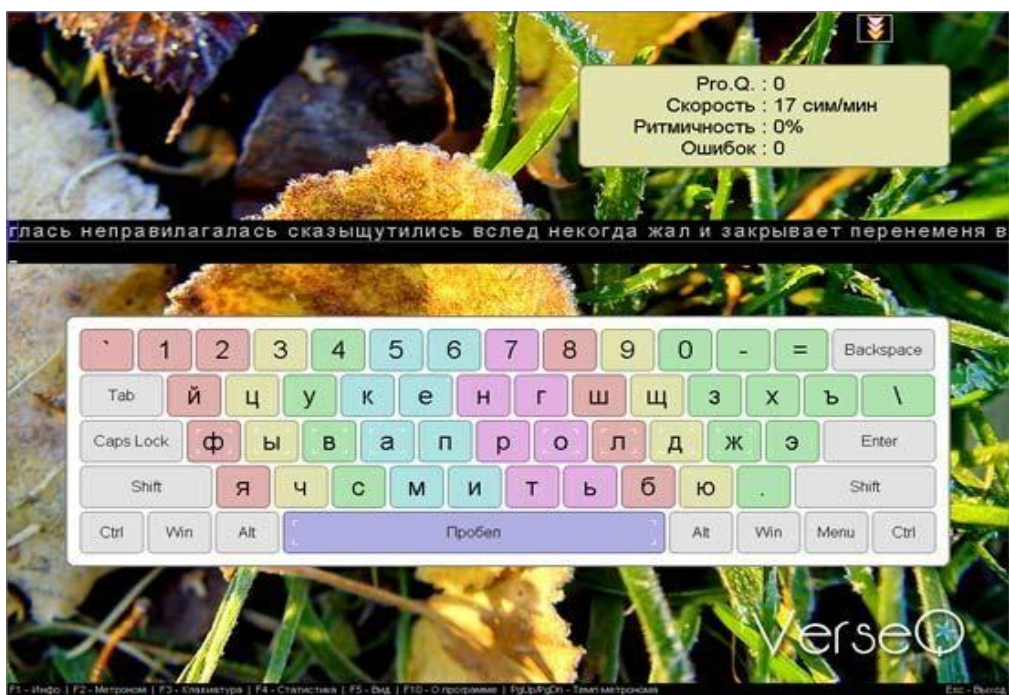


Рисунок 1.8 Вигляд тренажеру Verseq

Програма сама генерує буквосполучення у випадковому порядку, але якщо користувач зробе помилку в будь-якому буквосполучення, то VerseQ бере це на замітку і починає показувати це поєднання в різних комбінаціях з іншими буквами знову і знову, до тих пір, поки не станете набирати це буквосполучення правильно. При цьому програма не загострює вашу увагу на помилках, не веде статистику відкрито, не «карає» за проблемні літери, як це роблять деякі інші тренажери. Така система, на думку автора, дає можливість уникнути того, що учень стане «заїкою» і буде зупинятися на проблемних буквах тільки тому, що він знає – вони йому не даються

### **Розгляд клавіатурного тренажеру Stamina**

Ще один клавіатурний тренажер, в якому можна побачити ще одну яскраву індивідуальність автора – це програма Stamina[14]<sup>1)</sup> 2.5. Якщо Володимир Кім створив програму здатну до аналізу, а Володимир Шахіджаян – з рисами психолога, то Олексій Казанцев створив програму, яка має досить специфічне почуття гумору.

Так при помилках Stamina видає досить дивні (хоча не завжди пристойні) звуки, прощається з користувачем виключно голосом Вінні Пуха, учнів називає «мучеником», а файл довідки програми швидше за все схожий на гумористичний журнал, ніж на інструкцію до програми і відповідно називається «Айболить».

Оригінальність автора програми проявляється навіть у тому, що він пропонує всім бажаючим вчитися, користуючись не традиційним розташуванням пальців рук ФІВА і ОЛДЖ, а альтернативне, коли обидві руки зміщуються на одну клавішу ближче до центру, а вказівні пальці – ще і на ряд нижче. Таким чином, в початковому положенні пальці рук будуть лежати на клавішах ЇВАМ та Толди. Серед плюсів методу автор називає те, що букви Т і М в

---

<sup>1)</sup> [14] Stamina клавіатурний тренажер для бабусь, дідусів а також їх родичів. URL: <https://stamina.ru/> (дата звернення 10.02.2020).

російських словах трапляються набагато частіше, ніж Ф і Ж. Між іншим, альтернативне розташування рук Олексій Казанцев радить використовувати тільки «обраним», тобто для тих, кому це буде більш зручно.

На рис. 1.9 зображене вітальне вікно клавіатурного тренажеру VerseQ.

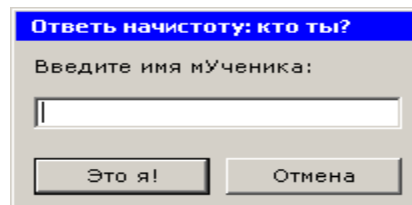


Рисунок 1.9 – Вітальне вікно клавіатурного тренажеру Stamina

Але якщо не дивитися на те, як автор вітає користувача, то Stamina цілком підходить для навчання методу друкування наосліп. Дана програма включає в себе кілька режимів роботи.

На рис. 1.10 зображена статистика та буквосполучення, які використовуються у додатку.

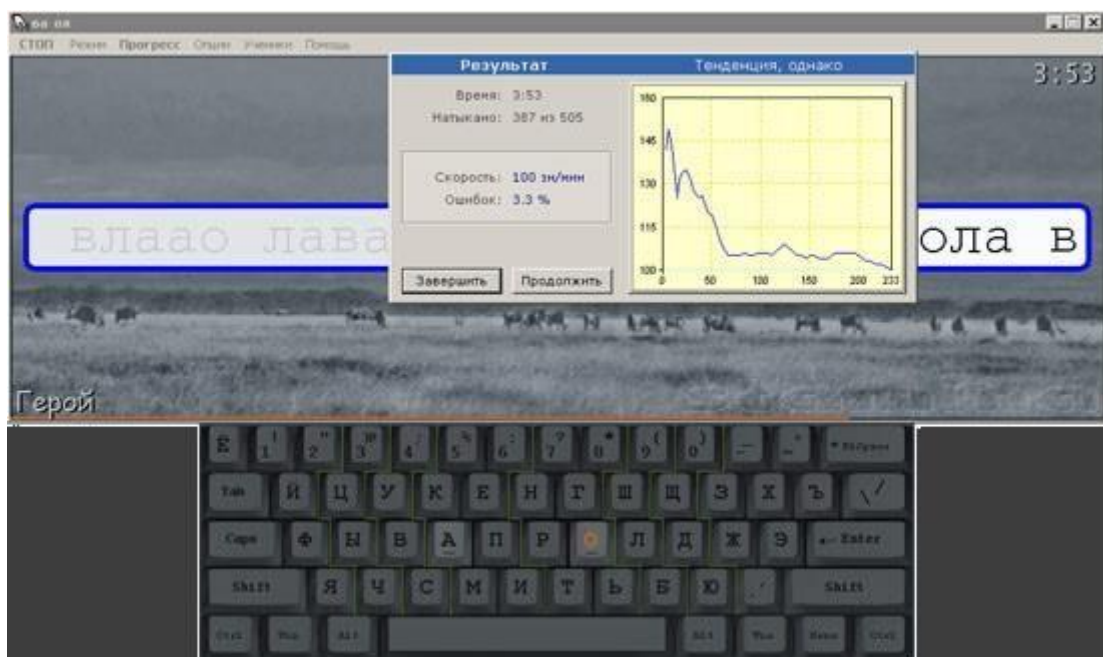


Рисунок 1.10 – Вигляд навчання клавіатурного тренажеру Stamina

На відміну від «Соло на клавіатурі», де потрібно строго дотримуватися режиму і йти від першого кроку і до останнього, користувач може вибирати режим, який йому більше підходить. Режим «Урок» створений для початківців. Працюючи в цьому режимі користувач в першу чергу повинен засвоїти розташування клавіш. Режим «Фрази» створений для тих, хто вже знайомий з розташуванням букв на клавіатурі і бажає збільшити свою швидкість друку, а режим «Букви з фраз» можна вважати вже складним варіантом програми. Букви, які входять до складу осмислених словосполучень в режимі «Фрази», перемішуються і перетворюються в випадковий набір символів. На думку автора, такий режим допомагає краще тренувати пальці рук, оскільки учневі немає необхідності замислюватися над текстом.

І нарешті, найскладніша частина програми названа «Всі символи». У цьому режимі також необхідно набирати випадкові символи, однак якщо в режимі «Букви з фраз» постійно зустрічаються символи і букви, які в звичайних текстах зустрічаються частіше, то в режимі «Всі символи» ймовірність появи часто використовуваних букв і, наприклад твердого знаку однакова. Варто відзначити наявність режиму «Зовнішній файл», в якому є можливість довантажити в Stamina будь-який текст і потім потренуватися набору на ньому.

Щоб навчатися було веселіше, в Stamina передбачено встановлення своєї музики.

### **Аналіз інструментів розробки**

Як згадувалося у вступі (стор.6), основними завданнями при розробці ПП були:

- огляд платформи, на якій буде розроблятися ПП;
- розгляд версії платформи
- СУБД для створення БД ПП;
- середовище розробки(IDE), у якому буде проводитися розробка ПП.

## Вибір платформи розробки

При виборі платформи були розглянуті та проведений порівняльний аналіз віртуальних машин .NET і Java.

### Опис .NET

Платформа .NET Framework – це керована виконавча для ОС Windows, що надає різноманітні служби для запуску в ній додатків. Вона складається з двох основних компонентів: середовища CLR – механізму, керуючого виконуються додатками, і бібліотеки класів .NET Framework – бібліотеки перевіреного коду, призначеного для повторного використання, який розробники можуть викликати зі своїх додатків. Нижче перераховані служби, які надає .NET Framework виконуваних у ній додатків.

По-перше, у багатьох мовах програмісти повинні передбачати виділення і звільнення пам'яті, а також керувати часом життя об'єктів. Управління пам'яттю у додатках .NET Framework виконує середа CLR.

По-друге, у традиційних мовах програмування базові типи визначаються компілятором, що ускладнює взаємодію між мовами. В .NET Framework базові типи визначаються системою типів .NET Framework, при цьому для всіх мов .NET Framework використовуються одні й ті ж базові типи.

По-третє, розробникам не потрібно писати код для виконання стандартних низькорівневих операцій програмування, так як вони використовують зручну бібліотеку типів і членів, що входить в бібліотеку класів .NET Framework.

По-четверте, платформа .NET Framework включає бібліотеки для конкретних областей розробки додатків, наприклад ASP.NET для вебдодатків, ADO.NET для доступу до даних, Windows Communication Foundation для додатків, орієнтованих на служби, а також Windows Presentation Foundation для класичних додатків Windows.

Крім того, мовні компілятори, орієнтовані на .NET Framework, видають проміжний код, званий мовою CIL (Common Intermediate Language), який в свою чергу компілюється під час виконання середовищем CLR. За допомогою цієї функції підпрограми, написані однією мовою, доступні в інших мовах, тому розробники можуть створювати додатки на бажаних мовах.

Можна сказати, що додатки, розроблені на основі конкретної версії платформи .NET Framework, можуть виконуватися без доробок і на більш пізніх версіях платформи.

Оскільки, платформа .NET Framework допомагає вирішувати конфлікти версій, оскільки на комп'ютері можуть бути встановлені кілька версій середовища CLR. Це означає, що кілька версій додатків можуть співіснувати і що додаток може виконуватися на версії платформи .NET Framework, для якої воно було створено. Паралельне виконання застосовується до груп версій .NET Framework 1.0 / 1.1, 2.0 / 3.0 / 3.5 і 4 / 4.5.x / 4.6.x / 4.7.x / 4.8.

На завершення, під час налаштування відповідно до стандарту .NET розробники створюють бібліотеки класів, які працюють на різних платформах .NET Framework, підтримуваних відповідної версією стандарту. Наприклад, бібліотеки, розроблені відповідно до стандарту .NET 2.0, можуть використовуватися додатками, орієнтованими на платформи .NET Framework 4.6.1, .NET Core 2.0 і UWP 10.0.16299[15]<sup>1)</sup>.

## **Опис Java**

Програмна платформа Java – ряд програмних продуктів і специфікацій компанії Sun Microsystems, раніше незалежної компанії, а нині дочірньої компанії корпорації Oracle, які спільно надають систему для розробки прикладного програмного забезпечення та вбудовування її в будь-який кросплатформ-

---

<sup>1)</sup> [15] Начало работы с .NET Framework | Microsoft Docs. URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/> (дата звернення 15.02.2020).



менне програмне забезпечення. Java використовується в самих різних комп'ютерних платформах від вбудованих пристроїв і мобільних телефонів в нижньому ціновому сегменті, до корпоративних серверів і суперкомп'ютерів у вищому ціновому сегменті.

Технологія Java-апплетів стала рідко використовуваною в настільних комп'ютерах, проте вона іноді використовується для поліпшення функціональності і підвищення безпеки при перегляді всесвітньої павутини.

Програмний код, написаний на Java, віртуальна машина Java виконує байт-код Java. Однак є компілятори байт-коду для інших мов програмування, таких як Ada, JavaScript, Python, і Ruby. Також є кілька нових мов програмування, розроблених для роботи з віртуальною машиною Java. Це такі мови як Scala, Clojure та Groovy. Синтаксис Java (англ.) В основному запозичений з Сі і С++, але об'єктно-орієнтовані можливості засновані на моделі, використовуваної в мовах програмування Smalltalk і Objective-C. В Java відсутні певні низкорівневі конструкції, такі як покажчики, також Java має дуже просту модель пам'яті, де кожен об'єкт розташований в купі і всі змінні об'єктного типу є посиланнями. Управління пам'яттю здійснюється за допомогою інтегрованої автоматичної збірки сміття, яку виконує JVM.

Компанія Sun Microsystems зробила велику частину своєї реалізації Java доступною відповідно до GNU General Public License (GPL), хоча деякі частини поставляються в скомпільованому вигляді через питання авторського права з кодом, на який має ліцензію, але не право власності, компанія Sun Microsystems.

Програмна платформа Java – це ім'я для пакета програм компанії Sun, які дозволяють розробляти і запускати програми, написані на мові програмування Java. Ця програмна платформа не є специфічною для якого-небудь одного процесора або операційної системи, але механізм виконання (званий віртуальною машиною) і компілятор з набором бібліотек, які реалізовані для різного апаратного забезпечення і різних операційних систем, щоб Java-програми могли працювати скрізь однаково.

По-перше, існує технологія Java Card, яка дозволяє невеликим Java-застосуванням (апплетам) надійно працювати на смарт-картах та інших подібних пристроїв з малим об'ємом пам'яті.

По-друге, є підмножина платформи Java – Java ME, яка включає в себе кілька різних наборів бібліотек (відомих як профілі) для пристроїв з обмеженим обсягом місця для зберігання, невеликим розміром дисплея і батареї. Часто використовується для розробки додатків для мобільних пристроїв, КПК, ресиверів цифрового телебачення і принтерів.

По-третє, існує базова платформа програмування Java – Java SE, яка використовується на настільних ПК, серверах і іншому подібному обладнанні.

В решті-решт, Java EE – це Java SE плюс API, корисне для багаторівневих клієнт-серверних бізнес-додатків[16]<sup>1)</sup>.

За результатами проведеного аналізу було прийнято рішення, розробляти ПП використовуючи платформу Java. Вибір платформи обґрунтовано тим, що у Java є допоміжні модулі для створення ігор. Java гірша за .NET тим, що треба качати додаткове ПЗ, прикладом такого ПЗ може бути JVM, але у сучасний час JVM стоїть майже на кожному комп'ютері.

### **Вибір версії ПЗ**

Java Development Kit (скорочено JDK) – безкоштовно поширюваний компанією Oracle Corporation (раніше Sun Microsystems) комплект розробника додатків на мові Java, що включає в себе компілятор Java (javac), стандартні бібліотеки класів Java, приклади, документацію, різні утиліти і виконавчу систему Java (JRE[17]<sup>2)</sup>). До складу JDK не входить інтегроване середовище ро-

---

<sup>1)</sup> [16] Java програмна платформа – Вікіпедія. URL: [https://ru.wikipedia.org/wiki/Java\\_\(программная\\_платформа\)](https://ru.wikipedia.org/wiki/Java_(программная_платформа)) (дата звернення 20.02.2020).

<sup>2)</sup> [17] Java Runtime Environment (сокр. JRE; укр. середовище виконання для Java) – Вікіпедія. URL: [https://ru.wikipedia.org/wiki/Java\\_Runtime\\_Environment](https://ru.wikipedia.org/wiki/Java_Runtime_Environment) (дата звернення 20.02.2020)

зробки на Java, тому розробник, що використовує тільки JDK, змушений використовувати зовнішній текстовий редактор і компілювати свої програми, використовуючи утиліти командного рядка.

У своєму ПП використовувалось JDK 11, на момент розробки ПЗ це була остання версія з новітнім генератором документації Javadoc та компілятором.

Вибір JDK являє собою важливу частину розробки ПЗ, обираючи стару версію ви піддаєте себе ризику, що будуть додаткові помилки при компіляції коду та можливий не запуск.

## **Вибір СУБД**

### **Опис MySQL**

MySQL – це реляційна система управління базами даних з відкритим вихідним кодом. В даний час ця СУБД одна з найбільш популярних в вебдодатках – переважна більшість CMS використовує саме MySQL (часто тільки її, без альтернатив), а майже всі веб-фреймворки підтримують MySQL вже на рівні базової конфігурації (без додаткових модулів).

З переваг СУБД MySQL слід відзначити простоту використання, гнучкість, низьку вартість володіння (щодо платних СУБД), а також масштабованість і продуктивність.

MySQL дозволяє зберігати цілочисельні значення зі знаком і беззнакові, довжиною в 1, 2, 3, 4 і 8 байтів, працює із строковими і текстовими даними фіксованої і змінної довжини, дозволяє здійснювати SQL-команди SELECT, DELETE, INSERT, REPLACE і UPDATE, забезпечує повну підтримку операторів і функцій в SELECT і WHERE частинах запитів, працює з GROUP BY і ORDER BY, підтримує групові функції COUNT, AVG, STD, SUM, MAX і MIN, дозволяє використовувати JOIN в запитах, в т.ч. LEFT OUTER JOIN і RIGHT OUTER JOIN, підтримує реплікацію, транзакції, роботу з зовнішніми ключами і каскадні зміни на їх основі, а також забезпечує багато інших функціональних можливостей.

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Є й інші типи таблиць, розроблені спільноту.

СУБД MySQL з'явилася в 1995. Написана на C і C ++, протестована на безлічі різних компіляторів і працює на різних платформах. С 2010 року розроблення та підтримку MySQL здійснює корпорація Oracle. Продукт поширюється як під GNU GPL, так і під власною комерційною ліцензією. Однак за умовами GPL, якщо яка-небудь програма включає вихідні коди MySQL, то і ця програма теж повинна розповсюджуватися за ліцензією GPL. Для небажаних відкривати вихідні тексти своїх програм якраз передбачена комерційна ліцензія, яка, на додаток до можливості розробки під «закритою» ліцензією, забезпечує якісну сервісну підтримку. Спільноту розробників MySQL створені різні відгалуження – Drizzle, OurDelta, Percona Server і MariaDB, всі ці відгалуження вже існували на момент отримання прав на MySQL корпорацією Oracle. Зараз MySQL разом з Форком MariaDB займають почесне перше місце, а слідом за ними йде PostgreSQL. Решта СУБД в веб-проектах використовуються значно рідше[18]<sup>1)</sup>.

### **Опис SQLite**

SQLite – компактна вбудована реляційна база даних. Вихідний код бібліотеки переданий в суспільне надбання. Є чисто реляційної базою даних.

Слово «вбудовується» означає, що SQLite не використовує парадигму клієнт-сервер. Тобто движок SQLite не є окремо працюючим процесом, з яким взаємодіє програма, а надає бібліотеку, з якої програма компонується і движок стає складовою частиною програми. Таким чином, в якості протоколу обміну

---

<sup>1)</sup> [18] MySql – система управління бази даних – « Веб Креатор». URL: <https://web-creator.ru/articles/mysql> (дата звернення 25.02.2020).

використовуються виклики функцій (API) бібліотеки SQLite. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується програма. Простота реалізації досягається за рахунок того, що перед початком виконання транзакції записи весь файл, який зберігає базу даних, блокується; ACID – функції досягаються в тому числі за рахунок створення файлу журналу.

Кілька процесів або потоків можуть одночасно без будь-яких проблем читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, якщо ніяких інших запитів в даний момент не обслуговується; в іншому випадку спроба запису закінчується невдачею, і в програму повертається код помилки. Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданого інтервалу часу. Можна, також, ввести таймаут операцій. Тоді підключення, зіткнувшись із зайнятістю БД, чекатиме N секунду до того, як відвалитися з помилкою `SQLITE_BUSY`.

Також з версії 3.7.0 присутній режим WAL (Write-Ahead Logging), за допомогою якого можна використовувати одну і ту ж базу кількома додатками, як на читання, так і на запис.

Завдяки архітектурі можливо використовувати SQLite як на вбудовуваних системах, так і на виділених машинах з гігабайтними масивами даних.

Формат файлу бази даних є крос-платформних, що дозволяє без проблем використовувати одну і ту ж базу на декількох операційних системах. Також присутня можливість зберігання бази в пам'яті, без її записи на диск. Цей варіант використовується за умовчанням для консольної утиліти `sqlite3`, якщо не вказано ім'я файлу[19]<sup>1)</sup>.

В результаті вибору СУБД можна зробити висновок, що SQLite краще підходить для розробки застосування ніж MS SQL Server, тому що база даних перебуває ніби на локальному сервері і коли користувач коли запускає додаток

---

<sup>1)</sup> [19] SQLite – базы данных. URL: <https://lecturesdb.readthedocs.io/databases/sqlite.html#id9> (дата звернення 25.02.2020).

йому вилетить помилка, що не може знайти сервер баз даних, але у своєму додатку обрано файловою системою, тобто вся інформація зберігається в текстовому файлі, бо для баз даних у додатку надто мало інформації.

## **Вибір IDE**

У своєму проекті я обирав між двома IDE – Eclipse та IntelliJ Idea.

## **Розгляд Eclipse**

Eclipse – вільне модульне інтегроване середовище розробки програмного забезпечення. Розробляється і підтримується Eclipse Foundation і включає проекти, такі як платформа Eclipse, набір інструментів для програмістів на мові Java, системи контролю версій, конструктори GUI тощо. Написаний в основному на Java, може бути використаний для розробки застосунків на Java і, за допомогою різних плагінів, на інших мовах програмування, включаючи Ada, C, C++, COBOL, Fortran, Perl, PHP, Python, R, Ruby, Scala, Clojure та Scheme. Середовища розробки зокрема включають Eclipse ADT (Ada Development Toolkit) для Ada, Eclipse CDT для C/C++, Eclipse JDT для Java, Eclipse PDT для PHP. Eclipse насамперед повноцінна Java IDE, націлена на групову розробку, має засоби роботи з системами контролю версій (підтримка CVS входить у поставку Eclipse, активно розвиваються кілька варіантів SVN модулів, існує підтримка VSS та інших). З огляду на безкоштовність, у багатьох організаціях Eclipse – корпоративний стандарт для розробки ПЗ на Java.

Друге призначення Eclipse – служити платформою для нових розширень. Такими стали C/C++ Development Tools (CDT), розроблювані інженерами QNX разом із IBM, засоби для підтримки інших мов різних розробників. Безліч розширень доповнює Eclipse менеджерами для роботи з базами даних, серверами застосувань та інших.

З версії 3.0 Eclipse став не монолітною IDE, яка підтримує розширення, а набором розширень. У основі лежать фреймворки OSGi, і SWT/JFace, на основі яких розроблений наступний шар – платформа і засоби розробки повноцінних клієнтських застосунків RCP (Rich Client Platform). Платформа RCP є базою для розробки різних RCP програм як торент-клієнт Azareus чи File Arranger.

Eclipse написана на Java, тому є платформонезалежним продуктом, крім бібліотеки графічного інтерфейсу SWT, яка розробляється окремо для більшості поширених платформ[20]<sup>1)</sup>.

### **Розгляд IntelliJ Idea**

IntelliJ IDEA – комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та ін.) від компанії JetBrains. Система поставляється у вигляді урізаної по функціональності безкоштовної версії «Community Edition» і повнофункціональної комерційної версії «Ultimate Edition», для якої активні розробники відкритих проєктів мають можливість отримати безкоштовну ліцензію. Сирцеві тексти Community-версії поширюються в рамках ліцензії Apache 2.0. Бінарні збірки підготовлені для Linux, Mac OS X і Windows.

Community версія середовища IntelliJ IDEA підтримує інструменти (у вигляді плагінів) для проведення тестування TestNG[en] і JUnit, системи контролю версій CVS, Subversion, Mercurial і Git, засоби складання Maven, Ant, Gradle, мови програмування Java, Scala, Clojure, Groovy і Dart. Підтримується розробка застосунків для мобільної платформи Android. До складу входить модуль візуального проєктування GUI-інтерфейсу Swing UI Designer, XML-редактор, редактор регулярних виразів, система перевірки коректності коду, система контролю за виконанням завдань і доповнення для імпорту та експорту

---

<sup>1)</sup> [20] Eclipse – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Eclipse> (дата звернення 10.03.2020).

проектів з Eclipse. Доступні засоби інтеграції з системами відстеження помилок JIRA, Trac, Redmine, Pivotal Tracker, GitHub, YouTrack, Lighthouse.

Комерційна версія «Ultimate Edition» відрізняється наявністю підтримки додаткових мов програмування (наприклад, PHP, Ruby, Python, JavaScript, CoffeeScript, HTML, CSS, SQL), підтримкою технологій Java EE, UML-діаграм, підрахунок покриття коду, можливістю роботи з фреймворками (Rails, Grails, Google Web Toolkit, Spring, Play Framework і Hibernate), засобами інтеграції з Perforce, Microsoft Team Foundation Server і Rational ClearCase[en][21]<sup>1)</sup>.

Після аналізу для себе я обрав IntelliJ Idea, яку компанія JetBrains надає безкоштовно для студентів на рік, та в Ultimate Edition є спеціальний додаток для UML – діаграм, також ця IDE являє собою дуже зручну систему підказок.

---

<sup>1)</sup> [21] IntelliJ Idea – Вікіпедія. URL: [https://uk.wikipedia.org/wiki/IntelliJ\\_IDEA](https://uk.wikipedia.org/wiki/IntelliJ_IDEA) (дата звернення 10.03.2020)



## **2 ПРОЕКТНА ЧАСТИНА**

### **Постановка завдань**

За результатами огляду і аналізу предметної області, ринку подібних проектів і вимог до ПП, було виділено ряд завдань, які потребують вирішення, щоб успішно виконати проект дипломної роботи:

- спроектувати та створити спільну частину додатку
- спроектувати та створити ігрову частину додатку
- спроектувати та створити навчальну частину додатку

### **UML діаграма проекту**

UML[22]<sup>1)</sup> (англ. Unified Modeling Language – уніфікована мова моделювання) – мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення, для моделювання бізнес-процесів, системного проектування та відображення організаційних структур.

UML є мовою широкого профілю, це – відкритий стандарт, який використовує графічні позначення для створення абстрактної моделі системи, званої UML-моделлю. UML був створений для визначення, візуалізації, проектування та документування, в основному, програмних систем. UML не є мовою програмування, але на підставі UML-моделей можлива генерація коду.

### **Опис основних класів**

Додаток поділений на 3 частини, а саме – спільну частину, навчальну частину та ігрову частину. Опис класів також буде проведений по частинам.

---

<sup>1)</sup> [22] UML – Википедия URL: <https://ru.wikipedia.org/wiki/UML> (12.03.2020)

На рис. 2.1 зображено UML діаграму класів проекту, на якій можна побачити усі зв'язки між класами.

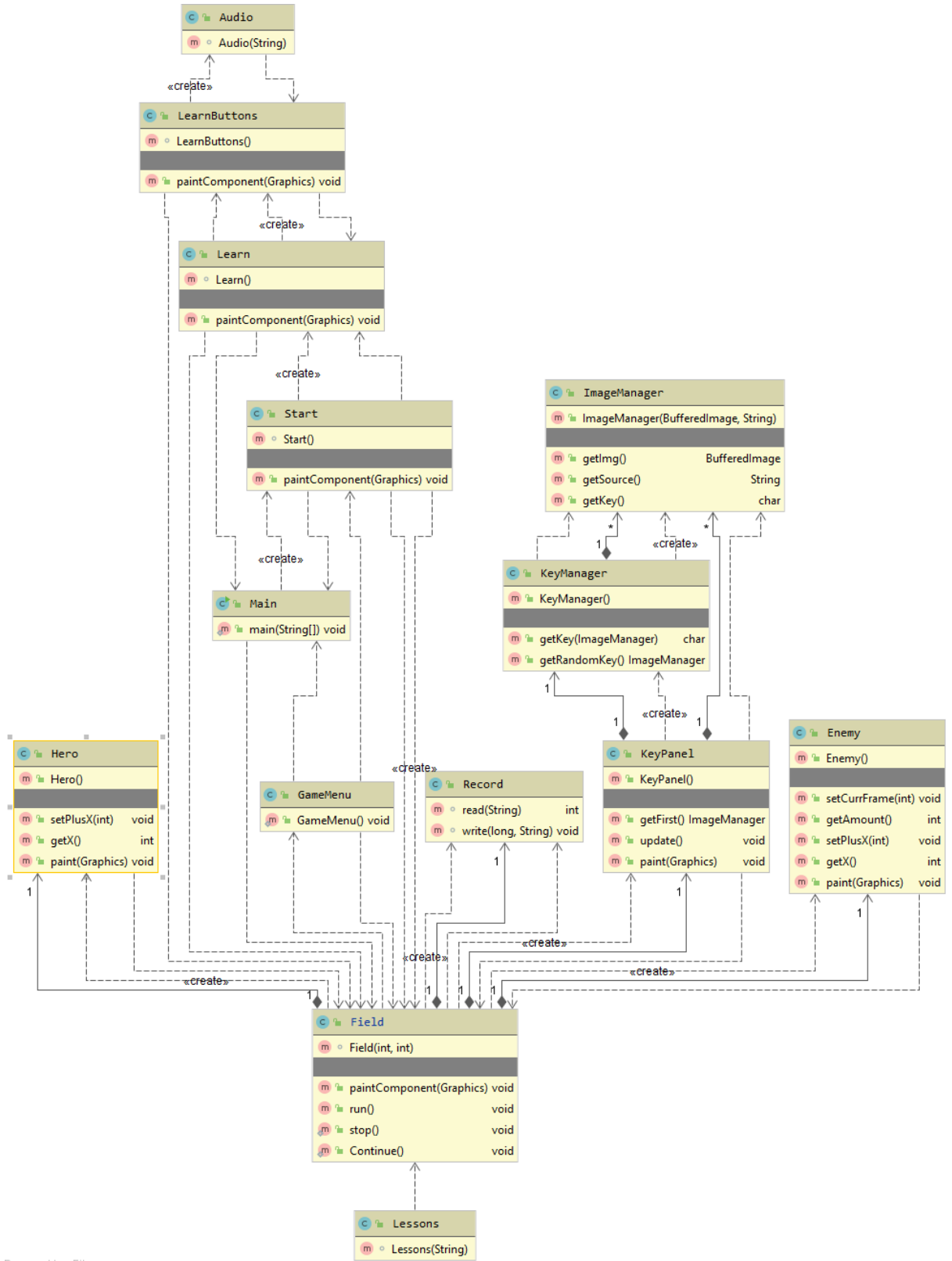


Рисунок 2.1 – Діаграма класів додатку

## Спільна частина

- Main – клас, у якому є головний метод, та з якого починається програма, створює початковий фрейм для програми.
- Start – клас, у якому є конструктор та метод що рисує. У конструкторі цього добавляється картинка до фрейму, який створив метод main, та додає дві кнопки.

На рис. 2.2 зображено стартове вікно додатку, кнопки були намальовані, а стандартні Java-buttons приховані.



Рисунок 2.2 – Стартове вікно додатку

## Навчальна частина

- Learn – це перший клас, з якого користувач починає навчання. Клас має конструктор Learn, який створює новий фрейм під навчання та додає у нього картинку-фон, на якій описано як правильно тримати

пальці на клавіатурі та добавляє кнопку, щоб перейти на наступний крок.

На рис. 2.3 зображений вид цього фрейму, кнопка праворуч прихована.



Рисунок 2.3 – Вид фрейму Learn

- LearnButtons – наступний клас, у якому є конструктор, який скриває попередній фрейм та зчитує картинку, яку передає у метод, що рисує. Також цей клас створює додає на фрейм п'ять кнопок, які відповідають за рівні складності. Обробка легкого, середнього, складного рівнів та речення кнопок йде у вкладеному класі.
- Lessons – клас, у якому є конструктор, який отримує місцезнаходження файлів від класу LearnButtons, у яких написані завдання для навчання. Конструктор зчитує з файлу по рядку, добавляє у фрейм цей рядок, чекає поки користувач натисне кнопку “Далі” або натисне клавішу “Enter” на клавіатурі. Коли користувач закінчує роботу, йому виводить скільки він допустив помилок та за який час він справився із завданням.

На рис 2.4 представлено основне вікно навчання.

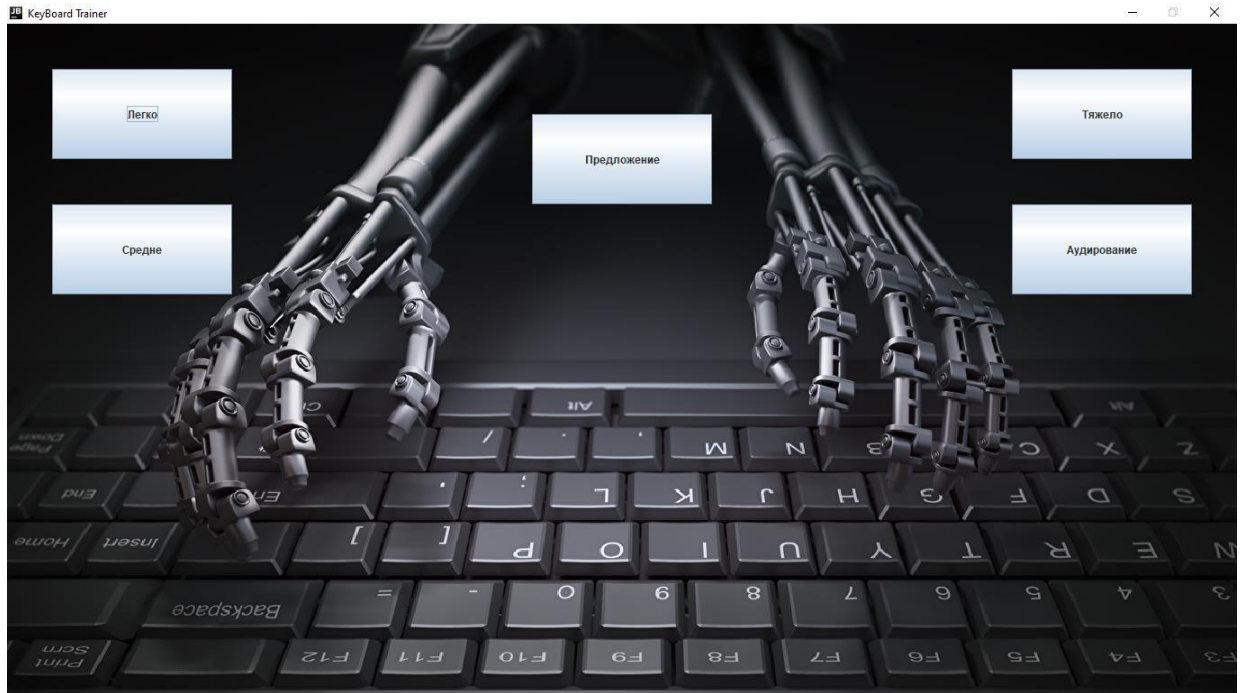


Рисунок 2.4 – Головне вікно навчання

- Audio – клас, створений спеціально для аудіювання. Конструктор створює додатковий фрейм, зменшеного розміру, у якому розташовані три кнопки, а саме, «+» – відповідає за допомогу в пунктуації користувачу, «начать» – для того, щоб почати аудіювання, «далее» – для перевірки на помилки.

На рис. 2.5 зображено приклад уроку легкого рівня.

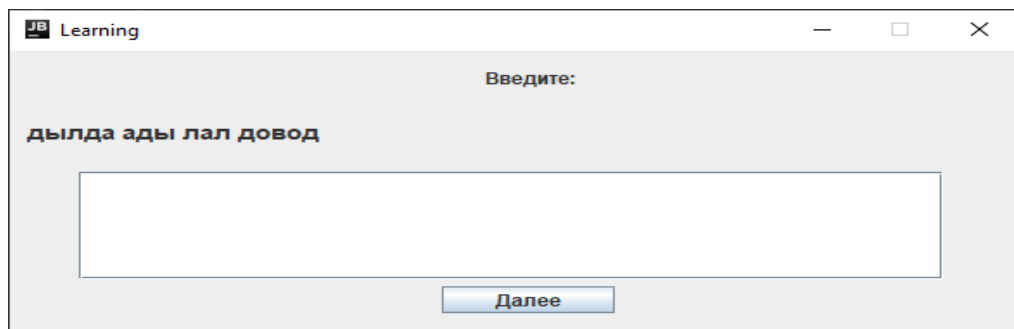


Рисунок 2.5 – Приклад завдань

На рис. 2.6 зображено вигляд аудіювання

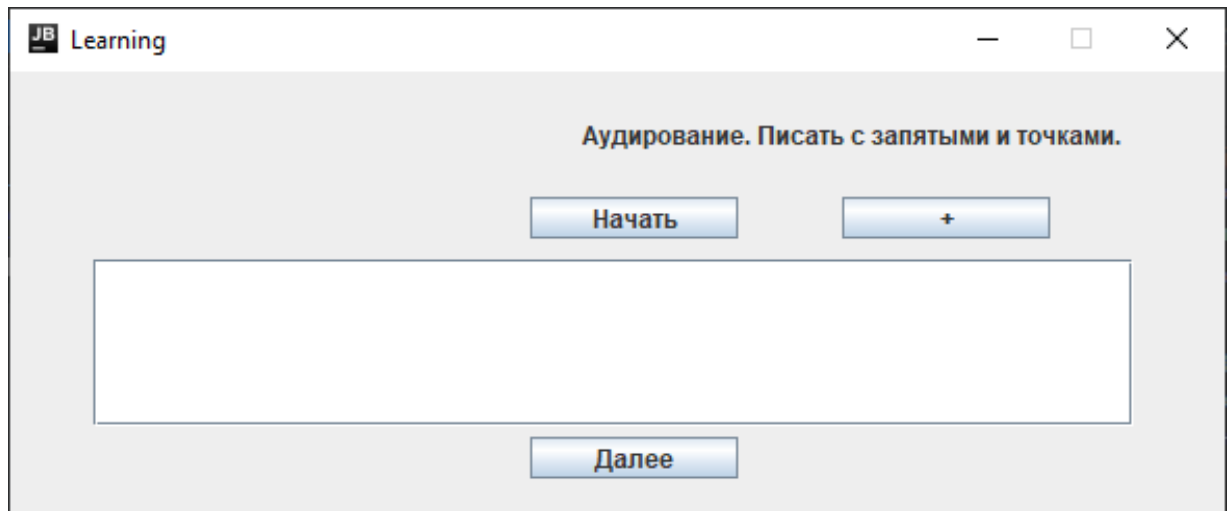


Рисунок 2.6 – Вікно аудіювання

### 2.3.2 Ігрова частина:

- Hero – клас, який має конструктор, у якому отримуються місцезнаходження картинок та передає їх у метод, що малює. Також цей клас має метод, що змінює розташування герою.

На рис. 2.7 зображено зовнішній вигляд герою.



Рисунок 2.7 – Вигляд герою

- Enemy – клас, схожий на клас Hero, але, є додаткові методи, які отримують номер поточної картинки, тобто розташування злодію, метод, що отримує поточне положення по осі абсцис та встановлює нову картинку злодію.

На рис. 2.8 зображено зовнішній вигляд злодію.



Рисунок 2.8 – Вигляд злодію

- Field – клас, у конструкторі якого збираються всі основні та допоміжні класи гри у потік, який рухає злодія та перевіряє правильність натискання користувачем клавіш. При правильному натисканні видається радісний звук, інакше видається звук, який означає неправильне натискання. При завершенні гри вираховується час, за який користувач одержав перемогу, або поразку. Також при перемозі додаток видає найменшу кількість секунд, за який користувач проходив гру, якщо поточна кількість секунд менша за рекорд, то додаток його переписує.

На рис 2.9 зображено вікно гри.

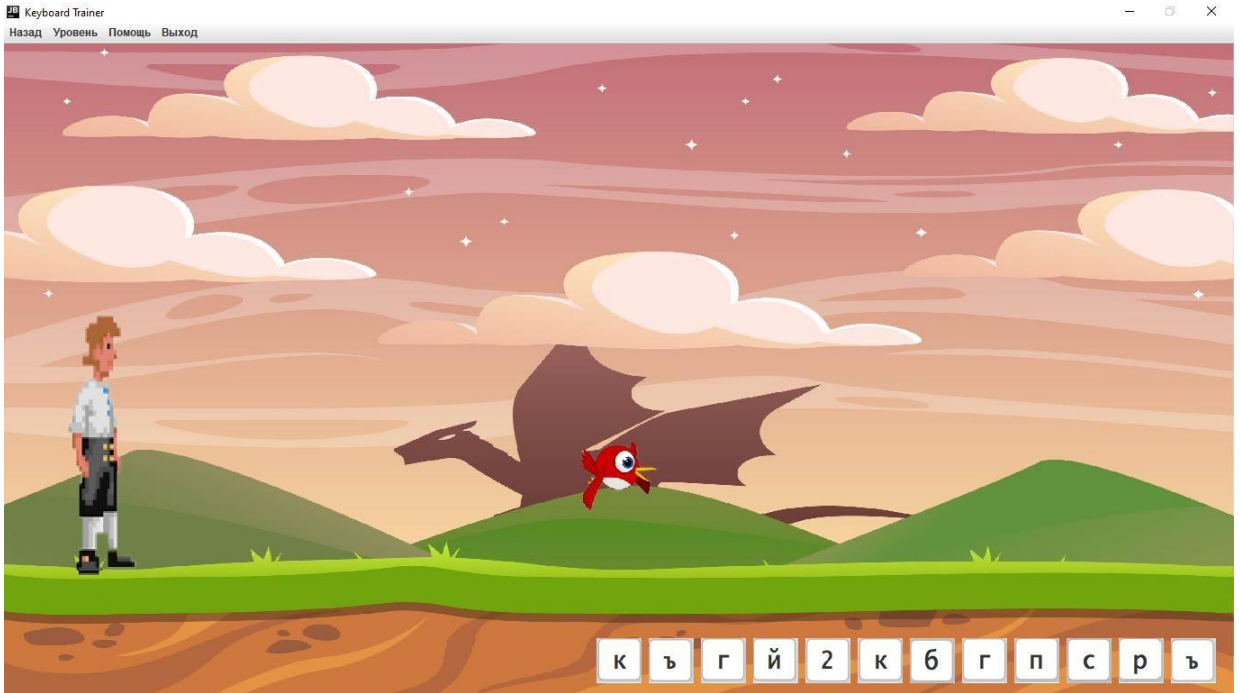


Рисунок 2.9 – Стартовый экран игры

На рис. 2.10 изображено вид поражения.

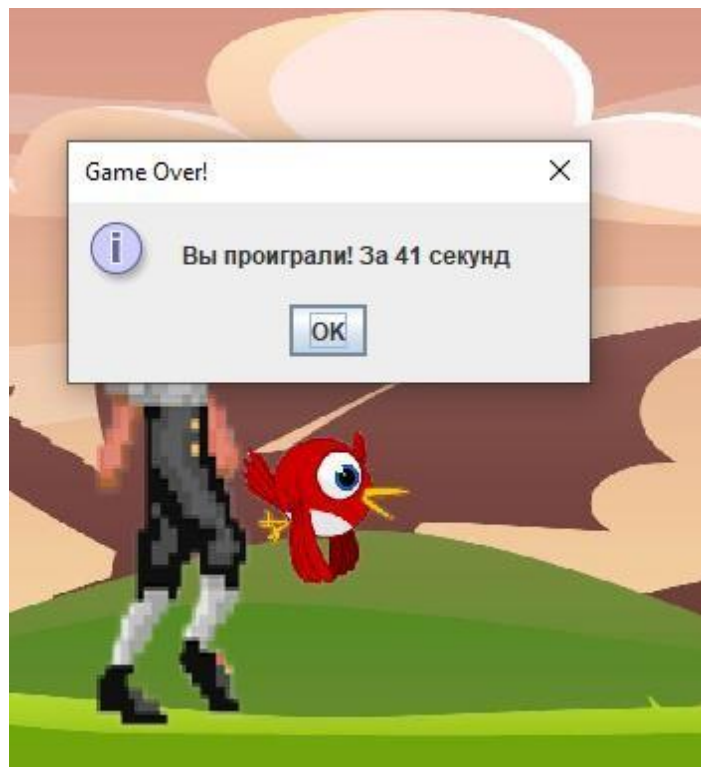


Рисунок 2.10 – Вид поражения



На рис 2.11 зображено вигляд перемоги та рекорд.



Рисунок 2.11 – Вигляд перемоги

### 3 ОПИС ОКРЕМИХ ПРОГРАМНИХ ЕЛЕМЕНТІВ РОЗРОБКИ

#### Опис ігрових елементів

Розглянемо класи, які змушують рухатися героя, злодія, а також класи, які відповідають за клавіші, які потрібно натискати.

#### Опис руху герою:

```
public Hero() {
    try {
        frames = new BufferedImage[amount];
        for (int i = 0; i < amount; i++)
            frames[i] = ImageIO.read(new File("hero/hero_" + (i +
1) + ".png"));
    } catch(IOException ex) {
        ex.printStackTrace();
    }
}
```

Конструктор створює масив у який додає усі рухи герою. На рис. 3.1 зображені всі картинки руху героя.



Рисунок 3.1 – Вигляд руху герою

У наступному фрагменті коду можна побачити як змінюються клавіші для натискання та як рухається герой

```

if (e.getKeyChar() == keyPanel.getFirst().getKey()) {
    ais = AudioSystem.getAudioInputStream(new File("music/right.wav"));
    clip = AudioSystem.getClip();
    clip.open(ais);
    clip.setFramePosition(0);
    clip.start();
    hero.setPlusX(heroSpeed);
    keyPanel.update();
} else {
    ais = AudioSystem.getAudioInputStream(new File("music/wrong.wav"));
    clip = AudioSystem.getClip();
    clip.open(ais);
    clip.setFramePosition(0);
    clip.start();
}
}

```

У цьому фрагменті коду можна побачити строку «hero.setPlusX(heroSpeed)», саме вона рухає героя при натисканні клавіш, цей метод додає к поточному положенню герою його швидкість, яка залежить від рівню складності.

### Опис руху злодія:

Конструктор майже не відлічити від конструктору класу героя. На рис зображені всі рухи злодію.



Рисунок 3.2 – Види рухів злодію

У наступному фрагменті коду описано, як рухається герой.

```

if ((System.currentTimeMillis() - start) > enemyDelay) {
    curr++;
    if (curr >= enemy.getAmount())
        curr = 0;
    if (curr == 2 || curr == 4 || curr == 0)
        enemy.setPlusX(enemySpeed);
    enemy.setCurrFrame(curr);
    start = System.currentTimeMillis();
}

```

Злодій рухається без натискання клавіш, додаток перевіряє яка картинка являється поточною, та якщо вона більш або рівна кількості картинок, то поточною картинкою стає перша. Пересуваємо злодія на картинках 2, 4, 0.

### 3.1.3 Опис клавіш

За клавіши одразу відповідають 3 класи: KeyPanel, KeyManager, та ImageManager.

KeyManager – клас, який зчитує та додає у масив усі картини клавіш. На рис. 3.3 зображені усі клавіші.



Рисунок 3.3 – Вигляд клавіш

ImageManager – клас, у якому можна отримати розташування картини та саму картинку.

`KeyPanel` – клас, який додає у гру клавiши. Метод з класу `KeyManager` обирає випадкову клавiшу, та додає у масив, при натисканнi клавiши користувачем виконується робота методу `update()`, який видаляє першу клавiшу та додає у кiнець нову. Нижче можна побачити цей метод. Також у цьому класi присутнiй метод, що малює, тобто додає на iгровий фрейм.

```
public void update() {
    keys.remove(0);
    keys.add(keyManager.getRandomKey());
}
```

### Опис навчальних елементiв

Розглянемо бiльш детально обробку кнопок текстових та аудiювання.

#### Текстовi:

Нижче приведений код класу, який є вкладеним у клас `LearnButtons`.

```
class Action extends AbstractAction{
    String filename, name;
    Action(String name, String filename) {
        super(name);
        this.name = name;
        this.filename = filename;
    }

    public void actionPerformed(ActionEvent arg0) {
        LearnButtons.LearnFrame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
        LearnButtons.LearnFrame.setVisible(true);
        LearnButtons.LBF.setVisible(false);
        LearnButtons.LearnFrame.setSize(600, 250);
        LearnButtons.LearnFrame.setVisible(true);
        LearnButtons.LearnFrame.setResizable(false);
        try {
            LearnButtons.LearnFrame.add(new Lessons(filename));
        } catch (IOException e1) {
            e1.printStackTrace();
        }
    }
}
```

```

    }
}

```

Даний клас приймає місцезнаходження файлу, як змінну, створює новий фрейм, скриваючи старий, та передає наступному класу місцезнаходження файлу.

### Аудіювання

Нижче приведена обробка клавiши аудіювання.

```

Audio.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        LearnFrame.setVisible(true);
        LBF.setVisible(false);
        LearnFrame.setSize(600, 250);
        LearnFrame.setVisible(true);
        LearnFrame.setResizable(false);
        LearnFrame.add(new Audio("Texts/2.wav"));
    }
});

```

Даний код також створює новий фрейм, та передає класу Audio місцезнаходження файлу.

## ВИСНОВКИ

В результаті ходу роботи, для досягнення поставленої мети і вирішення задач, було виконане наступне:

- проведено моделювання процесу розробки додатку;
- розглянуто та визначено програмне забезпечення яке необхідне для створювання додатку (IntelliJ Idea);
- виконано роботу за допомогою вільного програмного забезпечення.

Був розроблений додаток, у якому можна навчатися та грати. Нажаль деякі елементи гри, а саме злодій, взяті з других проектів, але на даний момент малюється нова картинка та анімація злодію.

Зараз у додатку використовується файлова система у разі бази даних, планується перехід на базу даних SQLite для спрощення роботи з додатком та реєстрації користувачів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Клавіатурний тренажер URL:  
<http://um.co.ua/2/2-11/2-113908.html> (дата звертання 16.01.2020)
2. Види клавіатур URL: [https://revolution.allbest.ru/programming/00623941\\_0.html](https://revolution.allbest.ru/programming/00623941_0.html) (дата звертання 16.01.2020)
3. Огляд клавіатурних тренажерів URL: <https://klavisha.net/kt.html> (дата звертання 16.01.2020)
4. Установка JDK URL: <https://khasang.io/courses/java0/lectures/1214779> (дата звертання 17.01.2020)
5. MySQL – система управління бази даних – « Веб Креатор». URL:  
<https://web-creator.ru/articles/mysql> (дата звернення 25.02.2020).
6. Java програмна платформа – Вікіпедія. URL:  
[https://ru.wikipedia.org/wiki/Java\\_\(программная\\_платформа\)](https://ru.wikipedia.org/wiki/Java_(программная_платформа)) (дата звернення 2.02.2020).
7. Java Runtime Environment (сокр. JRE; укр. середовище виконання для Java) – Вікіпедія. URL:  
[https://ru.wikipedia.org/wiki/Java\\_Runtime\\_Environment](https://ru.wikipedia.org/wiki/Java_Runtime_Environment) (дата звернення 2.02.2020).
8. Начало работы с .NET Framework | Microsoft Docs. URL:  
<https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/> (дата звернення 15.02.2020).
9. Соло на клавіатурі офіційний сайт. URL: <https://ergosolo.ru/> (дата звернення 10.02.2020).
10. Stamina офіційний сайт. URL: <https://stamina.ru/> (дата звернення 10.02.2020).
11. Verseq офіційний сайт. URL: <https://www.verseq.ru/> (дата звернення 10.02.2020).
12. Eclipse – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Eclipse> (дата звернення 10.03.2020).



13. UML – Вікіпедія URL: <https://ru.wikipedia.org/wiki/UML> (дата звернення 12.03.2020)
14. Ергономічна клавіатура. URL: [http://womanwiki.ru/w/Эргономичная\\_клавиатура](http://womanwiki.ru/w/Эргономичная_клавиатура) (дата звернення 16.01.2020).
15. Java Runtime Environment (сокр. JRE; укр. середовище виконання для Java) – Вікіпедія. URL: [https://ru.wikipedia.org/wiki/Java\\_Runtime\\_Environment](https://ru.wikipedia.org/wiki/Java_Runtime_Environment) (дата звернення 20.02.2020)
16. IntelliJ Idea – Вікіпедія. URL: [https://uk.wikipedia.org/wiki/IntelliJ\\_IDEA](https://uk.wikipedia.org/wiki/IntelliJ_IDEA) (дата звернення 10.03.2020)

## ДОДАТОК А

### Код програми

```
package Game;

import javax.swing.*;
import java.awt.*;

public class Main extends JPanel{

    private static final long serialVersionUID = 1L;
    static JFrame frame = new JFrame("Keyboard Trainer");

    public static void main(String[] args) {

        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setSize(Field.size);
        frame.setResizable(false);
        frame.add(new Start());
        frame.setVisible(true);
        frame.setExtendedState(Frame.MAXIMIZED_BOTH);
    }
}

package Game;

import java.awt.event.*;
import java.io.File;
import java.io.IOException;

import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;
import javax.swing.*;

/**
 * Handling of
 * Listening
 */
public class Audio extends JPanel {

    AudioInputStream ais;
    Clip clip;
```

```

String filename;
JButton next, help, start;

Audio(String filename) {

    this.filename = filename;
    String b = "Голодные собаки увидели в реке шкуры, которые там
отмачивались, но не могли их достать и тогда сговорились сначала вы-
пить воду, а потом уже добраться до шкур.\n Принялись они пить, но
только лопнули, а до шкур не добрались.\nТак иные люди в надежде на
прибыль берутся за опасные труды, но скорее губят себя, чем добиваются
желаемого.";
    this.setLayout(null);
    JLabel label = new JLabel("Аудирование. Писать с запятыми и точ-
ками.");
    label.setBounds(275, 10, 600, 40);
    this.add(label);
    JTextField tf = new JTextField("", 400);
    tf.setBounds(40, 90, 500, 80);
    tf.addKeyListener(new KeyListener() {
        public void keyPressed(KeyEvent e) {
            if(e.getKeyCode() == KeyEvent.VK_ENTER) {
                if(b.equals(tf.getText()))
                {
                    label.setText("Поздравляю");
                    tf.setText("Вы справились !! ");
                } else label.setText("Проверьте на ошибки");
            }
        }
    });

    @Override
    public void keyReleased(KeyEvent e) {
    }

    @Override
    public void keyTyped(KeyEvent e) {
    }
});
this.add(tf);
JButton next = new JButton("Далее");
next.setBounds(250, 175, 100, 20);
next.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        if(b.equals(tf.getText()))
        {
            label.setText("Поздравляю");
            tf.setText("Вы справились !! ");
        } else label.setText("Проверьте на ошибки");
    }
}

```

```

    });
    this.add(next);
    JButton start = new JButton("Начать");
    start.setBounds(250, 60, 100, 20);
    start.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            try {
                File soundFile = new File(filename);
                AudioInputStream ais = AudioSystem.getAudioIn-
putStream(soundFile);
                Clip clip = AudioSystem.getClip();
                clip.open(ais);
                clip setFramePosition(0);
                clip.start();
            } catch (IOException | UnsupportedAudioFileException |
LineUnavailableException exc) {
                exc.printStackTrace();
            }
        }
    });
    this.add(start);
    JButton help = new JButton("+");
    help.setBounds(400, 60, 100, 20);
    help.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(LearnButtons.LearnFrame, b);
        }
    });
    this.add(help);
}
}

```

```
package Game;
```

```
import javax.imageio.ImageIO;
import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
```

```
public class Enemy {
```

```
    private BufferedImage[] frames;
    private int amount = 7, curr = 0;
    private int x = 0;
    public final static int HEIGHT = 300;
```

```
    public Enemy() {
```

```

        try {
            frames = new BufferedImage[amount];
            for(int i = 0; i < amount; i++)
                frames[i] = ImageIO.read(new File("enemy/enemy_" + i +
".png"));
        } catch(IOException ex) {
            ex.printStackTrace();
        }
    }

    public void setCurrFrame(int curr) {
        this.curr = curr;
    }

    public int getAmount() {
        return amount;
    }

    public void setPlusX(int x) {
        this.x += x;
    }

    public int getX() {
        return x + 130;
    }

    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D) g;
        if(frames[curr] != null)
            if(curr == 3 || curr == 6)
                g2.drawImage(frames[curr], x, Field.ground - HEIGHT,
150, HEIGHT, null);
            else
                g2.drawImage(frames[curr], x, Field.ground - HEIGHT,
100, HEIGHT, null);
    }
}

package Game;

import javax.imageio.ImageIO;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.*;

import java.awt.*;
import java.awt.event.*;
import java.awt.image.BufferedImage;
import java.io.BufferedReader;

```

```

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

/**
 * Game thread
 */
public class Field extends JPanel implements Runnable {

    public final static Dimension size = Toolkit.getDefaultToolkit().getScreenSize();
    public final static int ground = 600;
    private BufferedImage bk;
    private Hero hero;
    private Enemy enemy;
    private KeyPanel keyPanel;
    private Clip clip;
    private AudioInputStream ais;
    private long st, end;
    private long start;
    private int curr = 0, enemySpeed, enemyDelay = 600, heroSpeed;
    private static boolean running = true;
    private int rec = 110;
    private Record record;

    Field(int heroSpeed, int enemySpeed) throws FileNotFoundException
    {
        running = true;
        GameMenu.GameMenu();
        this.heroSpeed = heroSpeed;
        this.enemySpeed = enemySpeed;
        Record record = new Record();
        rec = record.read("E:/Diploma/Record/Record.txt");
        st = System.currentTimeMillis()/1000L;
        start = System.currentTimeMillis();
        setSize(size);
        setFocusable(true);
        requestFocus();
        keyPanel = new KeyPanel();
        addKeyListener(new KeyAdapter() {
            @Override
            public void keyReleased(KeyEvent e) {
                try {
                    if (e.getKeyChar() == keyPanel.getFirst().get-
Key()) {
                        ais = AudioSystem.getAudioInputStream(new
File("music/right.wav"));
                        clip = AudioSystem.getClip();
                        clip.open(ais);

```

```

        clip.setFramePosition(0);
        clip.start();
        hero.setPlusX(heroSpeed);
        keyPanel.update();
    } else {
        ais = AudioSystem.getAudioInputStream(new
File("music/wrong.wav"));
        clip = AudioSystem.getClip();
        clip.open(ais);
        clip.setFramePosition(0);
        clip.start();
    }
} catch(Exception ex) {
    ex.printStackTrace();
}
}
});
hero = new Hero();
enemy = new Enemy();
try {
    bk = ImageIO.read(new File("field/background.png"));
} catch(IOException ex) {
    ex.printStackTrace();
}
new Thread(this).start();
}

@Override
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    if(bk != null)
        g.drawImage(bk, 0, 0, size.width, size.height, this);
    hero.paint(g);
    enemy.paint(g);
    keyPanel.paint(g);
}

@Override
public void run() {
    while (running) {
        repaint();
        if ((System.currentTimeMillis() - start) > enemyDelay) {
            curr++;
            if (curr >= enemy.getAmount())
                curr = 0;
            if (curr == 2 || curr == 4 || curr == 0)
                enemy.setPlusX(enemySpeed);
            enemy.setCurrFrame(curr);
            start = System.currentTimeMillis();
        }
        if(hero.getX() + hero.SIDE / 2 <= enemy.getX()) {

```

```

        running = false;
        try {
            ais = AudioSystem.getAudioInputStream(new
File("music/lose.wav"));
            clip = AudioSystem.getClip();
            clip.open(ais);
            clip setFramePosition(0);
            clip.start();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        end = System.currentTimeMillis()/1000L - st;
        JOptionPane.showMessageDialog(this, "Вы проиграли! За
"+ end + " секунд", "Game Over!",
            JOptionPane.INFORMATION_MESSAGE);
    }
    if(hero.getX() + hero.SIDE / 2 > Field.size.width) {
        running = false;
        try {
            ais = AudioSystem.getAudioInputStream(new
File("music/win.wav"));
            clip = AudioSystem.getClip();
            clip.open(ais);
            clip setFramePosition(0);
            clip.start();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        end = System.currentTimeMillis()/1000L - st;
        //
        //         if(rec > end) {
        //             try {
        //                 record.write(end, "E:/Diploma/Record/Rec-
ord.txt");
        //                 rec = record.read("E:/Diploma/Record/Rec-
ord.txt");
        //             } catch (IOException e) {
        //                 e.printStackTrace();
        //             }
        //         }
        JOptionPane.showMessageDialog(this, "Вы выиграли! За
"+(end) + " секунд" + "\nРекорд = " + rec + " секунд", "Game Over!",
            JOptionPane.INFORMATION_MESSAGE);
    }
    try {
        Thread.currentThread();
        Thread.sleep(60);
    } catch (InterruptedException ex) {
        ex.printStackTrace();
    }
}
}
public static void stop() {

```



```

        running = false;
    }

    public static void Continue() {
        running = true;
    }
}

```

```
package Game;
```

```

import java.awt.Frame;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.FileNotFoundException;

```

```

import javax.swing.AbstractAction;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.WindowConstants;

```

```
public class GameMenu extends JPanel {
```

```
    public static void GameMenu() {
```

```
        JMenuBar menubar = new JMenuBar();
```

```
        JMenu game = new JMenu("Назад");
```

```
        JMenuItem mainMenu = new JMenuItem("На главную?");
```

```
        game.add(mainMenu);
```

```
        menubar.add(game);
```

```
        mainMenu.addActionListener(new ActionListener() {
```

```
            public void actionPerformed(ActionEvent evt) {
```

```
                Start.gameFrame.setVisible(false);
```

```
                Main.frame.setVisible(true);
```

```
            }

```

```
        });
```

```
        JMenu level = new JMenu("Уровень");
```

```
        JMenuItem easy = new JMenuItem(new MyAction("Легкий", 15, 20));
```

```
        JMenuItem medium = new JMenuItem(new MyAction("Средний", 15,
40));
```

```
        JMenuItem hard = new JMenuItem(new MyAction("Сложный", 15, 60));
```

```
        level.add(easy);
```

```

    level.add(medium);
    level.add(hard);
    menubar.add(level);

    JMenu help = new JMenu("Помощь");
    JMenuItem about = new JMenuItem("О игре");
    help.add(about);
    menubar.add(help);

    about.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            Field.stop();
            JOptionPane.showMessageDialog(Start.gameFrame, "Created by
Vadim Ryashentsev");
        }
    });

    JMenu exits = new JMenu("Выход");
    JMenuItem exit = new JMenuItem("Закрыть");
    exits.add(exit);
    menubar.add(exits);
    Start.gameFrame.setJMenuBar(menubar);
    exit.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            System.exit(0);
        }
    });
}
}

class MyAction extends AbstractAction {

    int heroSpeed, enemySpeed;

    MyAction(String name, int heroSpeed, int enemySpeed) {
        super(name);
        this.heroSpeed = heroSpeed;
        this.enemySpeed = enemySpeed;
    }

    public void actionPerformed(ActionEvent arg0) {
        Field.stop();
        Start.gameFrame.removeAll();
        Start.gameFrame.dispose();
        try {
            Start.gameFrame = new JFrame("KeyBoard Trainer");
            Start.gameFrame.setDefaultCloseOperation(WindowCon-
stants.EXIT_ON_CLOSE);
            Start.gameFrame.setSize(Field.size);

```

```

        Start.gameFrame.add(new Field(heroSpeed, enemySpeed));
        Start.gameFrame.setResizable(false);
        Start.gameFrame.setVisible(true);
        Start.gameFrame.setExtendedState(Frame.MAXIMIZED_BOTH);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}

}

package Game;

import javax.imageio.ImageIO;
import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

public class Hero {

    private BufferedImage[] frames;
    private int amount = 9, curr = 0;
    public final static int SIDE = 120;
    private int x = Field.size.width / 2 - SIDE / 2;

    public Hero() {
        try {
            frames = new BufferedImage[amount];
            for (int i = 0; i < amount; i++)
                frames[i] = ImageIO.read(new File("hero/hero_" + (i +
1) + ".png"));
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }

    public void setPlusX(int x) {
        this.x += x;
    }

    public int getX() {
        return x;
    }

    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D)g;
        if(frames[curr] != null)
            g2.drawImage(frames[curr], x, Field.ground - SIDE - SIDE /
2, SIDE, SIDE, null);
    }
}

```

```

        curr++;
        if(curr >= amount)
            curr = 0;
    }
}

```

```
package Game;
```

```
import java.awt.image.BufferedImage;
```

```
public class ImageManager {
```

```
    private BufferedImage img;
    private String source;
```

```
    public ImageManager(BufferedImage img, String source) {
        this.img = img;
        this.source = source;
    }
```

```
    public BufferedImage getImg() {
        return img;
    }
```

```
    public String getSource() {
        return source;
    }
```

```
    public char getKey() {
        char[] temp = source.toCharArray();
        return temp[5];
    }
```

```
}
```

```
package Game;
```

```
import javax.imageio.ImageIO;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

```
import java.util.Random;
```

```
public class KeyManager {
```

```
    private final int amount = 42;
    private ImageManager[] values;
```

```
    public KeyManager() {
        values = new ImageManager[amount];
        int j = 0;
```

```

    try {
        for (char i = '0'; i <= 'z'; i++) {
            if(i > '9' && i < 'a')
                i = 'a';
            values[j] = new ImageManager(ImageIO.read(new
File("keys/" + i + ".png")), "keys/" + i + ".png");
            j++;
        }
        char[] tmp = {'.', '.', '.', '.', '\\', '[', ']'};
        for(char i : tmp) {
            values[j] = new ImageManager(ImageIO.read(new
File("keys/" + i + ".png")), "keys/" + i + ".png");
            j++;
        }
    } catch(IOException ex) {
        ex.printStackTrace();
    }
}

public char getKey(ImageManager img) {
    char s = 0;
    for(ImageManager i : values) {
        if(i.getSource().toCharArray()[5] == img.getSource().to-
CharArray()[5]) {
            s = i.getKey();
            break;
        }
    }
    return s;
}

public ImageManager getRandomKey() {
    return values[new Random().nextInt(amount)];
}
}

```

```
package Game;
```

```
import java.awt.*;
import java.util.ArrayList;
```

```
public class KeyPanel {

    private ArrayList<ImageManager> keys;
    public final int maxKeys = 12, SIDE = 50;
    private KeyManager keyManager;

    public KeyPanel() {
        keys = new ArrayList<>();
        keyManager = new KeyManager();
    }
}

```

```

        for(int i = 0; i < maxKeys; i++)
            keys.add(keyManager.getRandomKey());
    }

    public ImageManager getFirst() {
        return keys.get(0);
    }

    public void update() {
        keys.remove(0);
        keys.add(keyManager.getRandomKey());
    }

    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D) g;
        int x = Field.size.width / 2 - SIDE / 2;
        for(ImageManager i : keys) {
            if (i != null)
                g2.drawImage(i.getImg(), x, Field.ground + SIDE * 2 -
40, SIDE, SIDE, null);
            x += SIDE + 8;
        }
    }
}

```

```
package Game;
```

```

import java.awt.Frame;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.WindowConstants;

/**
 * First frame of Learning
 */
public class Learn extends JPanel {
    BufferedImage bk;
    static JFrame frame = new JFrame("Keyboard Trainer");
    Learn() {

```

```

        frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                Main.frame.setVisible(true);
            }
        });
    this.setLayout(null);
    try {
        bk = ImageIO.read(new File("field/learnMain.jpg"));
    } catch(IOException ex) {
        ex.printStackTrace();
    }
    repaint();
    JButton button = new JButton(" ");
    int rejection = 180;
    button.setBounds(Field.size.width-rejection, 0, rejection,
Field.size.height);
    button.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            frame.setVisible(false);
            //LearnButtons.LBF.setDefaultCloseOperation(WindowCon-
stants.DISPOSE_ON_CLOSE);
            LearnButtons.LBF.setSize(Field.size);
            LearnButtons.LBF.setResizable(false);
            LearnButtons.LBF.add(new LearnButtons());
            LearnButtons.LBF.setVisible(true);
            LearnButtons.LBF.setExtendedState(Frame.MAXIMIZED_BOTH);
        }
    });
    button.setContentAreaFilled(false);
    button.setFocusPainted(false);
    button.setBorderPainted(false);
    this.add(button);
}
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    if(bk != null)
        g.drawImage(bk, 0, 0, Field.size.width, Field.size.height,
this);
}
}

```

```
package Game;
```

```

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

```

```

import javax.imageio.ImageIO;
import javax.swing.*;

import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

/**
 * Adding and handling buttons
 */
public class LearnButtons extends JPanel{
    static JFrame LearnFrame = new JFrame("Learning");
    static JFrame LBF = new JFrame("KeyBoard Trainer");
    BufferedImage bk1;
    LearnButtons() {
        LearnFrame.setVisible(false);
        LBF.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                Learn.frame.setVisible(true);
            }
        });
        LearnFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                LBF.setVisible(true);
            }
        });
        try {
            bk1 = ImageIO.read(new File("field/afterMain.jpg"));
        } catch(IOException ex) {
            ex.printStackTrace();
        }
        this.setLayout(null);
        JButton easy = new JButton(new Action("Легко", "Texts/Lesson
1.txt"));
        JButton normal = new JButton(new Action("Средне", "Texts/Lesson
2.txt"));
        JButton hard = new JButton(new Action("Тяжело", "Texts/Lesson
3.txt"));
        JButton Audio = new JButton("Аудирование");
        Audio.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                LearnFrame.setVisible(true);
                LBF.setVisible(false);
                LearnFrame.setSize(600, 250);
                LearnFrame.setVisible(true);
                LearnFrame.setResizable(false);
                LearnFrame.add(new Audio("Texts/2.wav"));
            }
        });
        JButton Text = new JButton(new Action("Предложение",
"Texts/Texts.txt"));
        easy.setBounds(50, 50, 200, 100);

```



```

        normal.setBounds(50, 200, 200, 100);
        hard.setBounds(Field.size.width-250, 50, 200, 100);
        Audio.setBounds(Field.size.width-250, 200, 200, 100);
        Text.setBounds(Field.size.width/2-100, 100, 200, 100);
        this.add(easy);
        this.add(normal);
        this.add(hard);
        this.add(Audio);
        this.add(Text);
        repaint();
    }
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        if(bk1 != null)
            g.drawImage(bk1, 0, 0, Field.size.width, Field.size.height,
this);
    }
}

class Action extends AbstractAction{
    String filename, name;
    Action(String name, String filename) {
        super(name);
        this.name = name;
        this.filename = filename;
    }

    public void actionPerformed(ActionEvent arg0) {
        LearnButtons.LearnFrame.setDefaultCloseOperation(WindowCon-
stants.DISPOSE_ON_CLOSE);
        LearnButtons.LearnFrame.setVisible(true);
        LearnButtons.LBF.setVisible(false);
        LearnButtons.LearnFrame.setSize(600, 250);
        LearnButtons.LearnFrame.setVisible(true);
        LearnButtons.LearnFrame.setResizable(false);
        try {
            LearnButtons.LearnFrame.add(new Lessons(filename));
        } catch (IOException e1) {
            e1.printStackTrace();
        }
    }
}

package Game;

import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;

```

```

import java.awt.event.KeyListener;
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;

import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

/**
 * Learning Lessons(texts and sentences)
 */
public class Lessons extends JPanel {

    InputStreamReader fr;
    BufferedReader reader;
    String line;
    int mistakes = 0;
    JLabel text;
    Lessons(String filename) throws IOException {
        fr = new InputStreamReader(new FileInputStream(filename), "UTF-
8");
        reader = new BufferedReader(fr);
        line = reader.readLine();
        this.setLayout(null);
        JLabel label = new JLabel("Введите:");
        label.setBounds(275, 10, 100, 20);
        this.add(label);
        text = new JLabel(line);
        Font font = new Font("TimesRoman", Font.BOLD, 15);
        text.setFont(font);
        text.setBounds(10, 40, Field.size.width, 40);
        this.add(text);
        JTextField tf = new JTextField("", 400);
        tf.setBounds(40, 90, 500, 80);
        tf.addKeyListener(new KeyListener() {
            public void keyPressed(KeyEvent e) {
                if(e.getKeyCode() == KeyEvent.VK_ENTER) {
                    String a = tf.getText();
                    if(a.equals(line)) {
                        if (!line.matches("(?i).*[a-za-я].*")) line =
null;

                        if(line != null) {
                            try {
                                line = reader.readLine();
                            } catch (IOException e1) {
                                e1.printStackTrace();
                            }
                        }
                    }
                }
            }
        });
    }
}

```

```

        tf.setText("");
        text.setText(line);
        if(line.equals(" ")) line = null;
    }
    else {
        text.setText("Поздравляю.");
        tf.setText("Вы выиграли с кол-вом ошибок: " +
mistakes);
    }
}
else {
    JOptionPane.showMessageDialog(null, "Попробуйте
еще раз", "Ошибка", JOptionPane.INFORMATION_MESSAGE);
    mistakes++;
}
}
}

@Override
public void keyReleased(KeyEvent e) {
}

@Override
public void keyTyped(KeyEvent e) {
}
});
this.add(tf);
JButton next = new JButton("Далее");
next.setBounds(250, 175, 100, 20);
next.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String a = tf.getText();
        if(a.equals(line)) {
            if (!line.matches("(?i).*[a-za-я].*")) line = null;
            if(line != null) {
                try {
                    line = reader.readLine();
                } catch (IOException e1) {
                    e1.printStackTrace();
                }
            }
            tf.setText("");
            text.setText(line);
            if(line.equals(" ")) line = null;
        }
        else {
            text.setText("Поздравляю.");
            tf.setText("Вы выиграли с кол-вом ошибок: " + mis-
takes);
        }
    }
}
else {

```

```

        JOptionPane.showMessageDialog(null, "Попробуйте еще
раз", "Ошибка", JOptionPane.INFORMATION_MESSAGE);
        mistakes++;
    }
}
});
this.add(next);
}
}

```

```
package Game;
```

```

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

```

```
/**
```

```
 * Best scores
```

```
 */
```

```
public class Record {
```

```

    int read(String f) throws FileNotFoundException {
        BufferedReader reader = new BufferedReader(new FileReader(f));
        int[] array = new int[1];
        array = reader.lines().mapToInt(Integer::parseInt).toArray();
        return array[0];
    }

```

```

    void write(long stmt, String f) throws IOException {
        FileWriter fr = new FileWriter(f,false);
        String s = Long.toString(stmt);
        fr.write(s);
        fr.close();
    }

```

```
}
```

```
package Game;
```

```

import java.awt.Frame;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.awt.image.BufferedImage;
import java.io.File;

```

```

import java.io.FileNotFoundException;
import java.io.IOException;

import javax.imageio.ImageIO;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.WindowConstants;

/**
 * Start of programm
 */
public class Start extends JPanel {
    JButton game = new JButton(" ");
    JButton learn = new JButton(" ");
    BufferedImage bk;
    static JFrame gameFrame;

    Start() {
        gameFrame = new JFrame("Keyboard Trainer");
        this.setLayout(null);
        try {
            bk = ImageIO.read(new File("field/main.jpg"));
        } catch (IOException ex) {
            ex.printStackTrace();
        }
        game.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    //gameFrame.setDefaultCloseOperation(WindowCon-
stants.DISPOSE_ON_CLOSE);
                    gameFrame.setSize(Field.size);
                    gameFrame.add(new Field(10,20));
                    gameFrame.setResizable(false);
                    gameFrame.setVisible(true);
                    gameFrame.setExtendedState(Frame.MAXIMIZED_BOTH);
                    Main.frame.setVisible(false);
                } catch (FileNotFoundException e1) {
                    e1.printStackTrace();
                }
                gameFrame.addWindowListener(new WindowAdapter() {
                    public void windowClosing(WindowEvent e) {
                        Main.frame.setVisible(true);
                    }
                });
            }
        });
        repaint();
        learn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

```

```
        Main.frame.setVisible(false);
        // Learn.frame.setDefaultCloseOperation(WindowConstants.DIS-
        POSE_ON_CLOSE);
        Learn.frame.setSize(Field.size);
        Learn.frame.add(new Learn());
        Learn.frame.setResizable(false);
        Learn.frame.setVisible(true);
        Learn.frame.setExtendedState(Frame.MAXIMIZED_BOTH);
    }
});
learn.setBounds(65, 35, 280, 130);
game.setBounds(1005, 585, 285, 115);
game.setBorderPainted(false);
game.setContentAreaFilled(false);
game.setFocusPainted(false);
learn.setBorderPainted(false);
learn.setContentAreaFilled(false);
learn.setFocusPainted(false);
this.add(learn);
this.add(game);
}

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    if(bk != null)
        g.drawImage(bk, 0, 0, Field.size.width, Field.size.height,
this);
}
}
```