

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет _____ Магістерської та
_____ аспірантської підготовки

Кафедра інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Дослідження та розробка інтерактивних методів групового навчання програмуванню з використанням мобільних технологій»

Виконала студентка 2 року групи МК- 2
спеціальності 122 Комп'ютерні науки

_____ Каніковська Людмила Віталіївна

Керівник к.геогр.н., доц.

_____ Коваленко Людмила Борисівна

Консультант _____

Рецензент к.т.н., доцент

_____ Гнатовська Г.А.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук

Кафедра Інформаційних технологій

Рівень вищої освіти магістр

Спеціальність 122 Комп'ютерні науки

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

“ 29 ” жовтня _____ 2018 р.

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Каніковській Людмилі Віталіївні

(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження та розробка інтерактивних методів групового навчання програмуванню з використанням мобільних технологій»

керівник роботи к.т.н., доцент Трегубова Ірина Анатоліївна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ 5 ” жовтня 2018 р. №271-С

2. Строк подання студентом роботи 10.12.2018 р.

3. Вихідні дані до роботи мова програмування Java , середа розробки JetBrains Google Android Studio, засіб розробки інтерфейсу XML WYSIWYG editor, модулі Glide, Android-Iconics, MaterialDrawer, Gradle, Annotations, Google Android Support бібліотеки

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1 Аналіз існуючих систем для навчання програмуванню

1.1 Ресурс Udacity для вивчення програмування

1.2 Додаток Swift Playgrounds для вивчення мови програмування Swift

1.3 Ресурс Codecademy для вивчення програмування

1.4 Додаток Lrn для вивчення мов програмування та гіпер-розмітки

1.5 Порівняльний аналіз функціоналу систем для вивчення мов програмування

2 Огляд основних методологічних підходів навчання

2.1 Системний підхід у навчанні

2.2 Особистісний підхід у навчанні

2.3 Діяльнісний підхід у навчанні

2.4 Індивідуальний підхід

2.5 Тестування у навчальному процесі

2.6 Sandbox – рантайм компілятор коду

3 Опис розробленого програмного забезпечення та результатів

3.1 Концептуальне проектування бази даних інформаційної системи

3.2 Розробка додатку

5. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 29 жовтня 2018р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1	Огляд літературних джерел за темою магістерської роботи			
2	Аналіз основних методологічних підходів навчання			
3	Класифікація основних підходів навчання			
4	Рубіжна атестація	19.11.18-24.11.18		
5	Опис алгоритму створення тестових завдань з навчальної літератури			
6	Опис та аналіз структури розробленої бази даних			
7	Опис та аналіз виконаної роботи			
8	Оформлення висновків до роботи			
9	Здача роботи на кафедрі	10.12.18		
10	Перевірка на плагіат	13.12.18-14.12.18		
11	Рецензування	20.12.18		
	Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)			

Студент

(підпис)

Каніковська Л.В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Трегубова І.А.

(прізвище та ініціали)

АНОТАЦІЯ

на магістерську роботу

«Дослідження та розробка інтерактивних методів групового навчання програмуванню з використанням мобільних технологій»

Об'єкт дослідження – мобільні інформаційні технології і процес їх інтеграції в систему підтримки навчального процесу.

Предмет дослідження – методи проектування і розробки мобільних застосунків для навчання студентів мовам програмування.

Методи дослідження: об'єктно-орієнтоване моделювання, концептуальне моделювання.

Метою магістерської роботи є дослідження та розробка інтерактивних методів групового навчання програмуванню з використанням мобільних технологій..

Актуальність дослідження обумовлена необхідністю вирішення проблем підвищення якості освіти шляхом вдосконалення підходу до навчання студентів програмуванню з використанням мобільних технологій.

Результатом виконання магістерської роботи є розроблене мобільне застосування для вивчення мови програмування Java. Функціональність додатку передбачає можливість офлайн роботи, проходження різних типів тестів (Drag&Drop, множественний вибір, вибір однієї правильної відповіді), компіляції та запуску консольних додатків (режим Пісочниця), додавання статей до обраних.

Структура магістерської роботи складається з анотації, вступу, трьох розділів, висновків, переліку посилань на 21 найменувань. Повний обсяг роботи становить 65 сторінок і містить 19 рисунків.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, ТЕСТУВАННЯ, МОБІЛЬНИЙ ДОДАТОК, ПЛАТФОРМА ANDROID, РЕДАКТОР КОДУ JAVA, КОМПІЛЯТОР.

SUMMARY

for master's degree work

«Research and Development of Interactive Methods for Group Learning of Programming by Means of Mobile Technologies»

Object of research – mobile information technologies and the process of their integration into the system of educational process support.

Subject of research – methods of designing and developing mobile applications for teaching students in programming languages.

Research methods: object-oriented modeling, conceptual modeling.

The purpose of the master's thesis is to research and develop interactive methods of group learning for programming using mobile technologies.

The relevance of the study is due to the need to address the problems of improving the quality of education by improving the approach to teaching students with the use of mobile technologies.

The result of the master's thesis is the developed mobile application for learning the Java programming language. The functionality of the application includes the possibility of offline work, passage of various types of tests (Drag & Drop, multiple choice, the choice of one correct answer), compilation and run console applications (Sandbox mode), adding articles to favorites.

The structure of the master's thesis consists of an abstract, an introduction, three chapters, conclusions, a list of references to 21 titles. The full volume of work is 65 pages and contains 19 figures.

Keywords: INFORMATION SYSTEM, TESTING, MOBILE APPLICATION, ANDROID PLATFORM, JAVA CODE EDITOR, COMPILER.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
1 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ДЛЯ НАВЧАННЯ ПРОГРАМУВАННЮ.....	11
1.1 Ресурс Udacity для вивчення програмування	11
1.2 Додаток Swift Playgrounds для вивчення мови програмування Swift	12
1.3 Ресурс Codecademy для вивчення програмування	13
1.4 Додаток Lrn для вивчення мов програмування та гіпер-розмітки.....	14
1.5 Додаток Tynker для вивчення алгоритмізації та програмування.....	15
1.6 Ресурс Coursera для онлайн навчання	16
1.7 Ресурс Khan Academy для онлайн навчання.....	17
1.8 Ресурс EdX для онлайн навчання.....	18
1.9 Ресурс CodeCombat для онлайн навчання.....	20
1.10 Ресурс CodeWars для онлайн навчання	21
1.11 Порівняльний аналіз функціоналу систем для вивчення мов програмування	22
2 ОГЛЯД ОСНОВНИХ МЕТОДОЛОГІЧНИХ ПІДХОДІВ НАВЧАННЯ.....	24
2.1 Системний підхід у навчанні.....	24
2.2 Особистісний підхід у навчанні	24
2.3 Діяльнісний підхід у навчанні.....	25
2.4 Індивідуальний підхід	25
2.4.1 Візуальне сприйняття інформації	27
2.4.2 Дотикове (тактильне) сприйняття інформації	29
2.5 Тестування у навчальному процесі.....	31
2.5.1 Методи генерації тестових завдань	32
2.5.2 Drag-and-drop технологія	40
2.6 Sandbox – рантайм компілятор коду.....	43
3 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА РЕЗУЛЬТАТІВ	45
3.1 Структура розробленого програмного забезпечення.....	45
3.1.1 Support – бібліотеки підтримки	45
3.1.2 Модуль Android Annotations	47
3.1.3 Система компіляції та збірки проекту Gradle	47
3.1.4 Модуль бокового меню MaterialDrawer	48
3.1.5 Модуль підтримки векторних іконочних шрифтів Android-Iconics	49
3.1.6 Модуль завантаження та обробки зображень Glide	49
3.1.7 Особливості Android Studio	50
3.2 Концептуальне проектування бази даних інформаційної системи	52
3.3 Розробка додатку	56

ВИСНОВКИ	63
ПЕРЕЛІК ПОСИЛАНЬ	64
ДОДАТОК А. ЖИТТЄВИЙ ЦИКЛ ANDROID ФРАГМЕНТУ	Ошибка! Закладка не определена.
ДОДАТОК Б. ЖИТТЄВИЙ ЦИКЛ ANDROID ДОДАТКУ	Ошибка! Закладка не определена.
ДОДАТОК В. ПРИКЛАД ПРОГРАМНОГО КОДУ З ВИКОРИСТАННЯМ ANDROID ANNOTATIONS	Ошибка! Закладка не определена.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМИНІВ

Скорочення

БД	– база даних;
БЗ	– база знань;
ІС	– інформаційна система;
КПК	– кишеньковий персональний комп'ютер;
НФ	– нормальна форма;
ООП	– об'єктно-орієнтоване програмування;
ПЗ	– програмне забезпечення;
ПТМ	– тезисова концептуальна модель;
СКБД	– система керування базами даних;
ТЗ	– технічне завдання;
API	– Application Programming Interface (інтерфейс прикладного програмування);
CMS	– Content management system (Система управління контентом);
CSS	– Cascading Style Sheets (Каскадні таблиці стилів);
DTD	– Document Type Declaration (Декларація типу документу);
ERD	– Entity-Relationship Diagrams (Діаграми сутність-зв'язок);
HTML	– HyperText Markup Language (Мова розмітки гіпертекстових документів);
MVC	– Model-View-Controller (Модель-вид-контролер);
SQL	– Structured Query Language (Мова структурованих запитів);
UCS	– Universal Character Set (Універсальний Набір Символів);
UML	– Unified Modeling Language (Універсальна мова моделювання);
URL	– Uniform Resource Locator (Єдиний вказівник ресурсу);

ВСТУП

В наш час людям доводиться обробляти велику кількість інформації, тому вони вимушені шукать способи економії часу. Якщо поррахувати, то в черзі та в транспорті ми витрачаємо багато часу, який могли би потратити на щось більш важливе. Сучасному студенту неможливо впоратися з колосальним потоком інформації, який щодня обрушується на нього. Навіть високорозвинена пам'ять не в змозі зберегти гігантський інформаційний масив. Тому з'являються нові завдання в навчанні, пов'язані з чітким відбором навчального матеріалу, структуруванням загальноосвітнього курсу програмування, складання удосконалених методик вивчення, що дозволяють за малу кількість часу отримувати максимум інформації.

Мета магістерської роботи – дослідження та розробка інтерактивних методів групового навчання програмуванню з використанням мобільних технологій. В роботі передбачається розробка веб-додатка для вивчення мов програмування.

Актуальність роботи обумовлена тим, що в наш час, коли смартфони є у кожній людини, мобільні технології вже активно використовуються при вивченні, наприклад, іноземних мов, тому є сенс вивчати таким чином і мови програмування.

Грамотна реалізація подібного веб-застосування надасть можливість студентам молодших курсів університету ознайомитися з основами програмування та отримати можливість самопідготовки до контролюючих засобів з дисципліни «Алгоритмізація та програмування». Ця програма може бути одним з елементів вдосконалення освітнього процесу підготовки молодших спеціалістів з ІТ в коледжах та технікумах. Реалізація зручного інтерфейсу веб-застосування зробить процес навчання більш простим та зрозумілим для початківців програмістів.

Для досягнення поставленої мети в роботі необхідно вирішити наступні завдання:

- виконати аналіз подібних існуючих мобільних систем для вивчення мов програмування;
- обґрунтувати вибір методів, що будуть використовуватися для вирішення проблеми ефективного вивчення програмування;
- надати методологічні основи вирішення поставленої проблеми;
- виконати проектування програмного забезпечення;

- розробити алгоритми та програмні реалізації веб-додатку для вивчення мов програмування;
- виконати тестування розробленого програмного забезпечення;
 - виконати аналіз отриманих результатів.

1 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ДЛЯ НАВЧАННЯ ПРОГРАМУВАННЮ

1.1 Ресурс Udacity для вивчення програмування

Udacity – відомий ресурс для навчання програмуванню – випустив безкоштовний додаток, завдяки якому користувач може істотно підвищити рівень своєї освіти, проходячи безкоштовні курси з програмування від провідних експертів ІТ-галузі (рис.1.1). Доступний для Android та iOS [1].

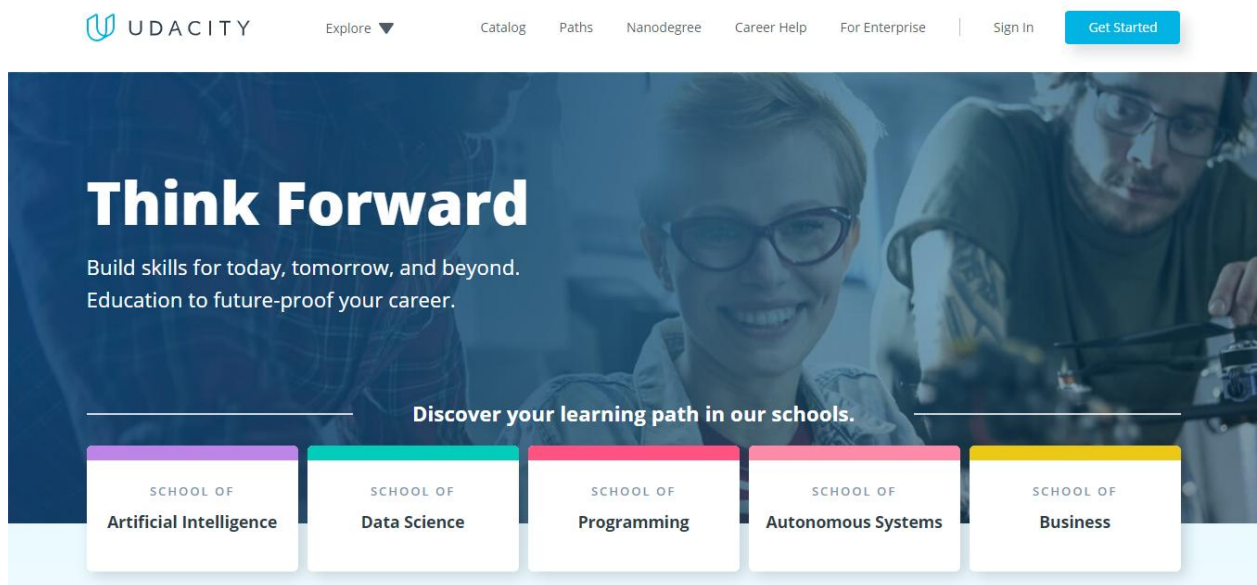


Рисунок 1.1 – Головна сторінка ресурсу Udacity

Udacity – приватна освітня організація, заснована Себастьяном Трунов, Девідом Ставенсом і Майклом Сокольський, з метою демократизації освіти. Компанія виникла в результаті розширення програми з інформатики Стенфордського університету. Дистанційні курси доступні безкоштовно через Інтернет, прослухати їх може будь-хто. Спочатку пропонувалося шість курсів. Станом на 1 жовтня 2012 року Udacity пропонує 14 курсів. Число студентів становить десятки тисяч людей. Про заснування Udacity було оголошено в 2012 році на конференції Digital Life Design. У 2012 році Себастьян Трун був відзначений газетою The Guardian як людина, що вніс істотний внесок в розвиток відкритого Інтернету.

Відео лекції англійською мовою з субтитрами в поєднанні з вбудованими тестами і подальшими домашніми роботами, засновані на моделі «вчитися

на практиці». Кожна лекція включає в себе вбудований тест, щоб допомогти студентам зрозуміти пропоновані концепції та ідеї.

До штучного інтелекту, робототехніки та фізики додалися курси з ведення блогів і створення стартапів, але не все йде гладко. Багато ідей провалилися, як, наприклад, курс з дискретної математики – рівень виявився занадто низьким порівняно з іншими. З іншого боку, на форумах досі багато незадоволених: курси, на їхню думку, перебувають за межею можливостей і штучно ускладнені. Команда проекту приймає це до відома – простіше, звичайно, не буде, але способи викладання матеріалу і тести постійно переглядаються, щоб якомога більше студентів змогли освоїти предмет, незважаючи на складність.

1.2 Додаток Swift Playgrounds для вивчення мови програмування Swift

Swift Playgrounds – революційний додаток для iPad, щоб навчитися та експериментувати з кодом (рис.1.2). Користувач вирішує інтерактивні головоломки на уроках "Керування кодом", щоб опанувати основи кодування, тоді як додаткові проблеми дозволяють вивчати код і створювати програми, які є привабливими та унікальними [2].

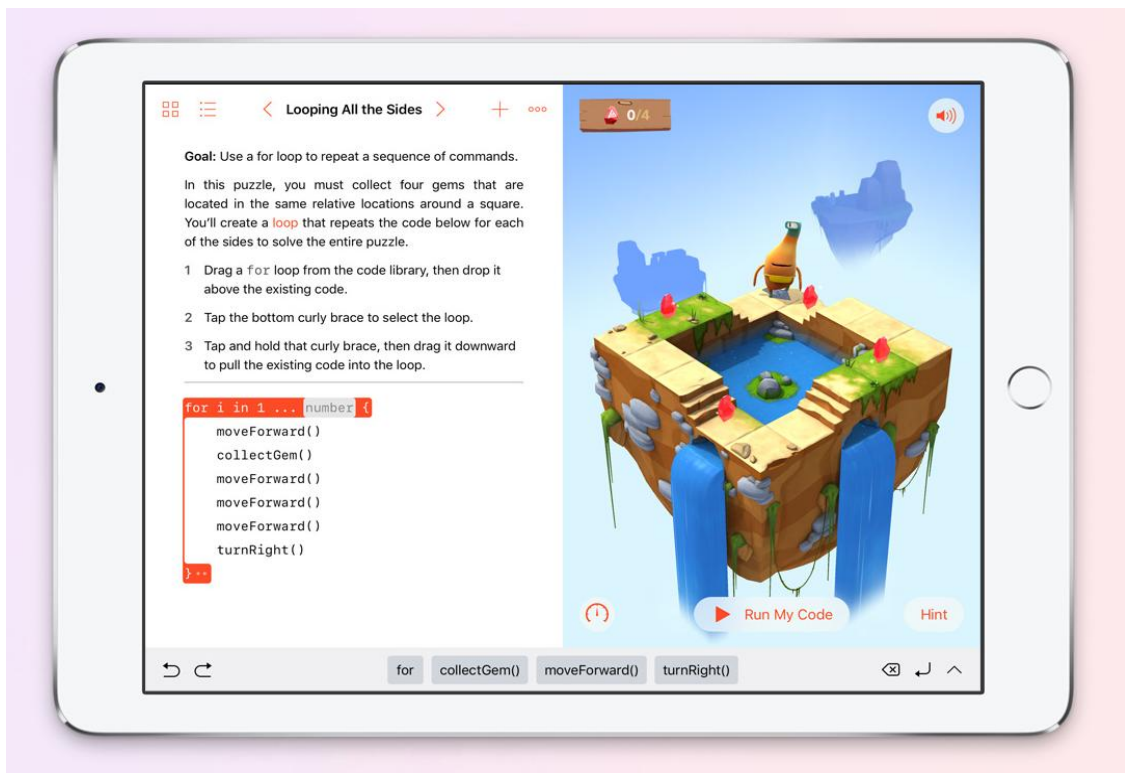


Рисунок 1.2 – Скріншот Swift Playgrounds

Swift Playgrounds не вимагає знань кодування, так що це ідеально підходить для студентів, які тільки починають. Весь час вони можуть вивчати Swift – потужну мову програмування, створену Apple, яка використовується професіоналами для створення багатьох популярних програм сьогоднішнього дня. І тому, що він створений для повного використання iPad та справжнього SDK для iOS, Swift Playgrounds є першим в своєму роді навчальним досвідом.

Вбудовані уроки Apple допоможуть: проаналізувати основні концепції програмування, використовуючи код для вирішення головоломок; переглядати свій код у красивому, інтерактивному 3D-вигляді, який можна обертати за допомогою пальця для збільшення; анімації впроваджують кожен нову концепцію кодування на високому рівні, перш ніж зануритися у головоломки; вибрати один з трьох анімованих символів, щоб виконати дії коду; використовувати глосарій, що охоплює повний перелік загальних термінів, треба лише торкнутися слова в інструкціях для отримання додаткової довідки.

1.3 Ресурс Codecademy для вивчення програмування

Codecademy – інтерактивна онлайн-платформа навчання 12 мовами програмування (рис.1.3): Python, PHP, JavaScript, Ruby, Java та ін., а також робота з jQuery бібліотекою та мовами розмітки та оформлення веб-сторінок HTML та CSS [3]. За станом на січень 2014 року, 24 мільйони користувачів виконали більше 100 мільйонів вправ. Codecademy отримала безліч позитивних відгуків від багатьох блогів і сайтів, зокрема New York Times і TechCrunch.

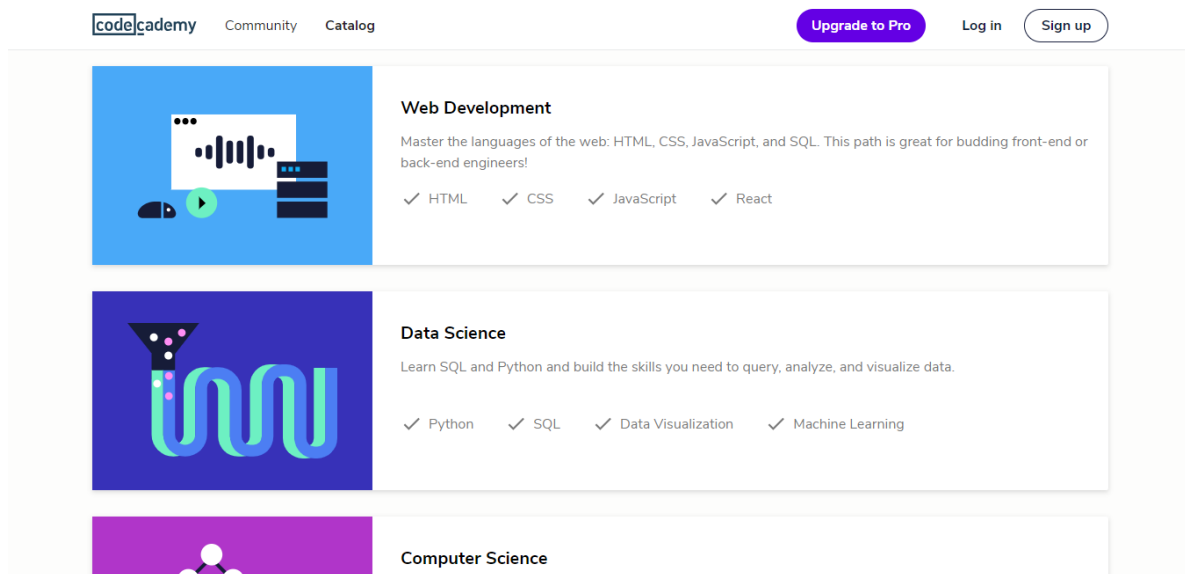


Рисунок 1.3 – Сторінка з розділами навчання ресурсу Codecademy

Кожен користувач має свій власний профіль. Для мотивації користувачів існує система сприятливих досягнень за виконанням навчань, індикатор курсу прогресу, який може побачити інші користувачі сайту. Так само доступні словники HTML та CSS в межах одного курсу. Сайт дозволяє кожному створювати і публікувати нові курси, використовуючи Творця курсу. На Codecademy також існує форум, де початківці та досвідчені веб-розробники можуть обмінюватися та допомагати один одному. Для деяких курсів існують «песочниці», в яких користувачі можуть тестувати свої програмні коди.

Прийняв участь в Тиждні Освітньої Інформатики в грудні 2013 року, Codecademy запустила свій перший iOS-додаток «Година Кодексу». Примітка фокусується на основи програмування і розроблена для людей, які хочуть вивчати програмування в ігровому вигляді.

1.4 Додаток Lrn для вивчення мов програмування та гіпер-розмітки

Lrn – це сервіс, який допоможе навчитись писати код однією з популярних мов програмування: HTML, CSS, Python, Ruby, Javascript (рис.1.4). Доступний у вигляді додатка для iOS (Android-версія в розробці) [4].

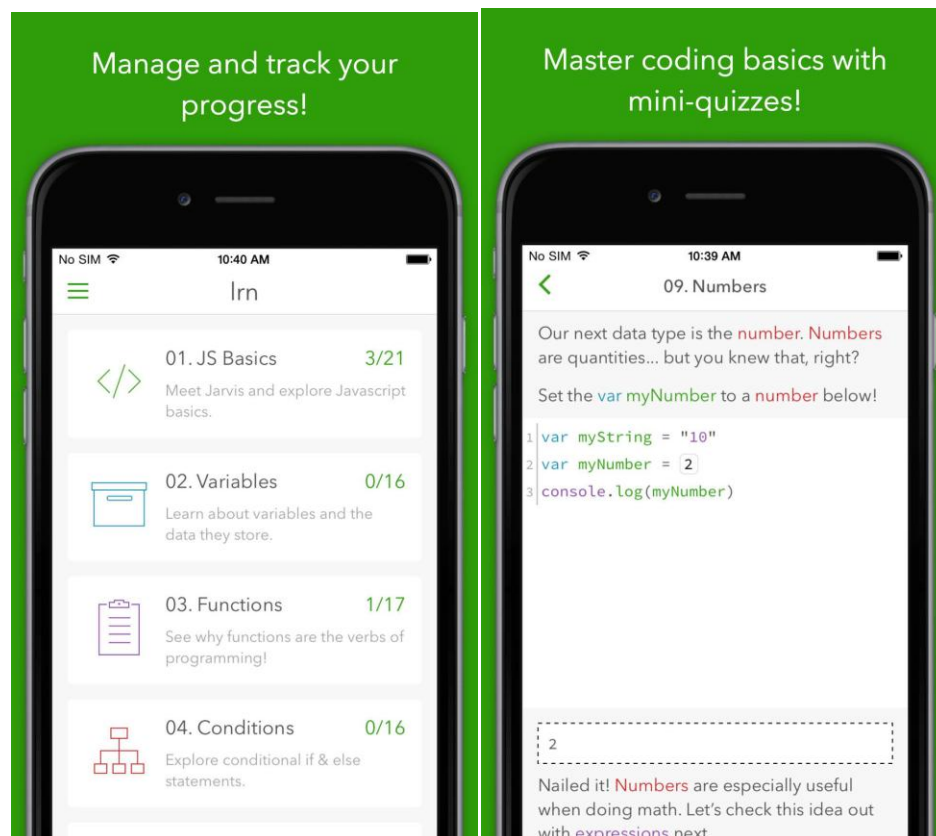


Рисунок 1.4 – Скріншот додатка Lrn

Додаток допомагає навчитися кодувати в HTML, CSS, Javascript, Ruby та Python за допомогою інтерактивних міні-тестів. Користувач отримує розуміння того, як читати, писати та говорити на мові програмування. Ніякого досвіду, Інтернету чи клавіатури не потрібно.

У додатку можуть бути розблоковані 400 + безкоштовні міні-вікторини та ще 200 + міні-вікторини. Курс складається з наступних глав: курс HTML: HTML інтро, списки, таблиці, форми, макет; курс CSS: CSS інтро, селектори, властивості, проміжок часу, позиціонування; курс Javascript: основи JS, змінні, функції, умови, петлі, масиви, масиви II, об'єкти, об'єкти II; курс Ruby: Ruby інтро, змінні, методи, умови, петлі, колекції, класи; курс Python: основи PY, змінні, функції, умови, списки, словники, петлі.

1.5 Додаток Tynker для вивчення алгоритмізації та програмування

Tynker – одна з найпопулярніших навчальних платформ для дітей (рис.1.5). Вона використовується більше як у 8000 шкіл і допомагає понад 6 млн дітям почати програмувати. Також доступний iOS-додаток.

Компанія Tynker була заснована в 2012 році в Маунтін-В'ю, штат Каліфорнія, США. З фондами, залученими від ангелів та інституційних інвесторів, Tynker for Schools розпочався в квітні 2013 року, а "Tynker for Home" став наступним роком.

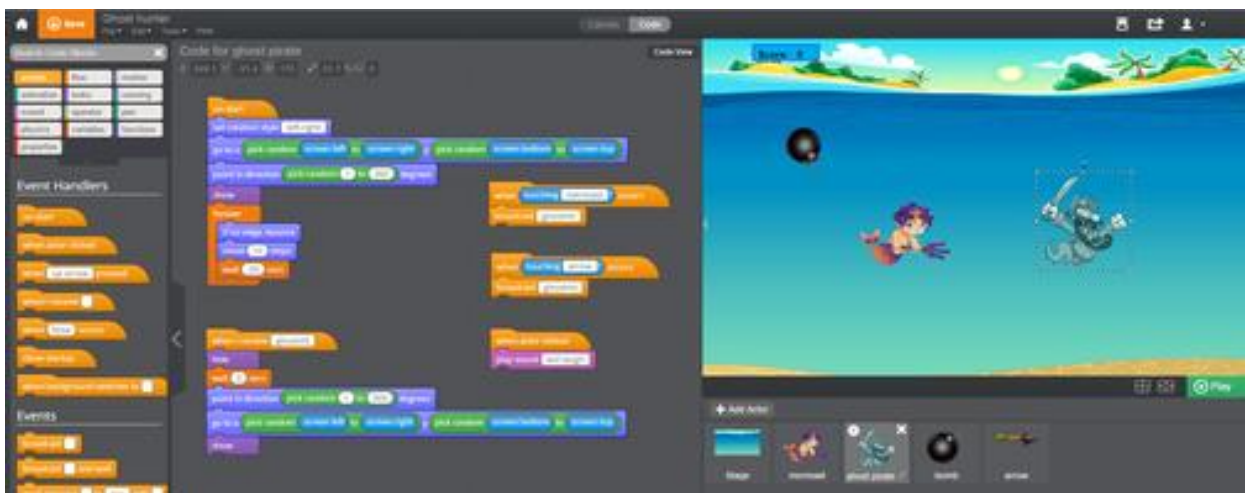


Рисунок 1.5 – Скріншот програми Tynker

Tynker – платформа освітнього програмування, спрямована на навчання дітей, як робити ігри та програми. Замість того, щоб вводити вихідний код,

можна візуально перетягувати блоки коду. Візуальний дизайн та принципи засновані на безкоштовній Scratch, як і Hopscotch та Snap. На відміну від Scratch, Tynker не заснований на власних Adobe Flash, але HTML5 і JavaScript, і може використовуватися в браузері без плагінів, а також на планшетах і смартфонах. Інша відмінність полягає в тому, що Scratch – це безкоштовний проект із відкритим кодом, а Tynker – комерційний продукт, призначений для продажу курсів.

У липні 2014 року Tynker був випущений для iPad та Android. Проекти можна отримати з Інтернету та планшета, і вони використовуються на будь-якій платформі. Станом на 28 лютого 2015 року ці платформи ще не сумісні з 100%, а деякі функції доступні лише на одній платформі.

1.6 Ресурс Coursera для онлайн навчання

Coursera – це онлайн-платформа для навчання, заснована професорами Стенфордського університету Ендрю Нг і Дафні Коллером, яка пропонує курси, спеціалізації та ступені (рис.1.6). Coursera співпрацює з університетами та іншими організаціями з надання онлайн-курсів, спеціалізацій та ступенів з різних предметів, таких як інженерія, гуманітарні науки, медицина, біологія, соціальні науки, математика, бізнес, інформатика, цифровий маркетинг, інформатика та інші. Станом на червень 2018 року в "Курсе" було більше 33 мільйонів зареєстрованих користувачів та більше 2400 курсів [6].

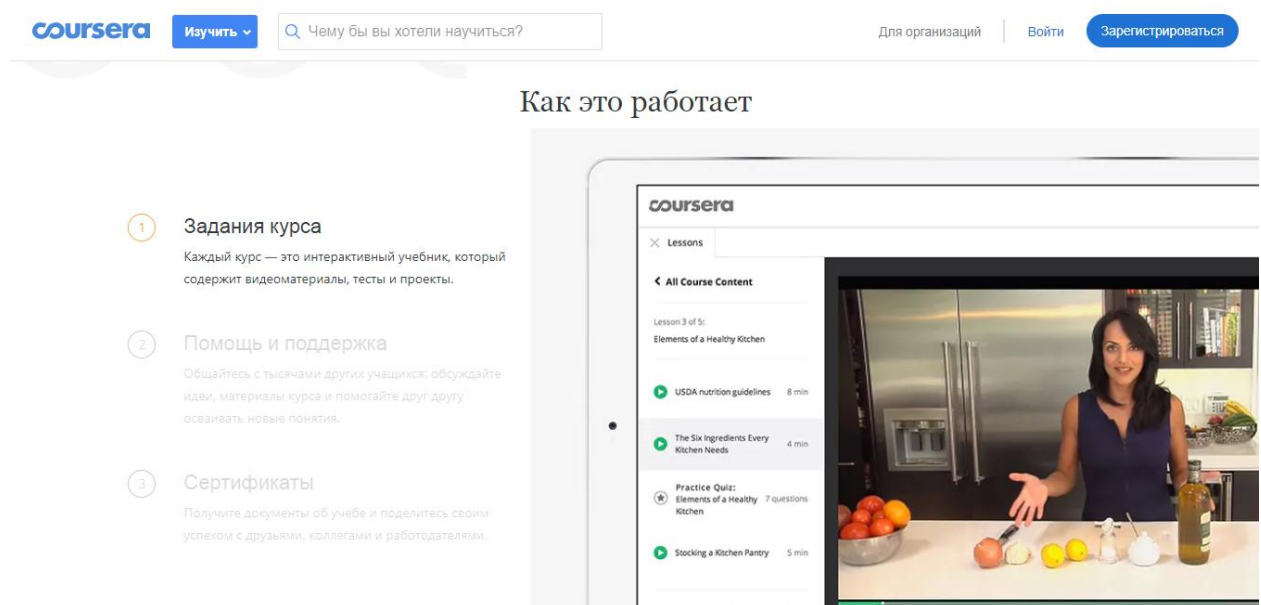


Рисунок 1.6 – Головна сторінка ресурсу Coursera

Coursera була заснована в 2012 році професорами інформатики Стенфордського університету Ендрю Нг та Дафні Коллером. Н.Г. і Коллер були натхнені своїм досвідом, пропонуючи свої курси Стенфорда в Інтернеті восени 2011 року, і незабаром після того, як він залишив Стенфорд, щоб запустити "Курсера". Принстон, Стенфорд, Мічиганський університет та Пенсильванський університет були першими університетами, які пропонують вміст на платформі. Пропозиції з моменту розширення включають спеціалізації – збірки курсів, які складають навички у конкретній темі, а також ступені та продукти розробки робочої сили для бізнесу та державних організацій [5].

Початковий обсяг фінансування проекту у розмірі 16 мільйонів доларів за підтримки Kleiner Perkins Caufield & Byers та New Enterprise Associates. У 2013 році Група Світового банку очолювала інвестиції в серію В, що становила 63 мільйони доларів. У 2015 році компанія EDB Investments очолила серію С венчурного фінансування, яка становила понад 60 мільйонів доларів. На сьогоднішній день компанія отримала фінансування на суму 146,1 млн. Доларів.

1.7 Ресурс Khan Academy для онлайн навчання

Khan Academy, відомий освітній ресурс, випустив додатки для Android та iOS, які дають можливість проходити курси платформи, де завгодно (рис.1.7).





Math	Science & engineering	Computing	Arts & humanities	Economics & finance	Test prep	College, careers, & more
 Math		Early math Algebra 1 Trigonometry AP Calculus AB Differential equations		Arithmetic Geometry Precalculus AP Calculus BC Linear algebra	Pre-algebra Algebra 2 Statistics & probability Multivariable calculus	
 Math by grade		Kindergarten 3rd grade 6th grade Illustrative Mathematics	1st grade 4th grade 7th grade Eureka Math/EngageNY		2nd grade 5th grade 8th grade High school	
 Science & engineering		Physics Cosmology & astronomy Organic chemistry AP Biology		AP Physics 1 Chemistry Biology Health & medicine	AP Physics 2 AP Chemistry High school biology Electrical engineering	
 Computing		Computer programming Computer animation		Computer science	Hour of Code	

Рисунок 1.7 – Головна сторінка ресурсу Хан Академії

Khan Academy є неприбутковою освітньою організацією, створеною в 2008 році Салманом Хан, з метою створення набору Інтернет-інструментів, які допомагають навчати студентів [7]. Організація випускає короткі уроки у формі відео YouTube. Її веб-сайт також включає в себе додаткові практичні заняття та матеріали для вчителів. Всі ресурси доступні для користувачів веб-сайту. Веб-сайт та його вміст надаються переважно англійською мовою, але також доступні іншими мовами, включаючи іспанську, португальську, іврит, італійську, російську, китайську, турецьку, французьку, бенгальську, хінді та німецьку мови.

Організація розпочалася у 2008 році, коли Салман Хан навчав одного з його двоюрідних братів у галузі математики в Інтернеті, використовуючи службу під назвою Yahoo! Doodle Images. Через деякий час інші двоюрідні брати хана почали користуватися його репетиторською службою. Завдяки попиту, Хан вирішив зробити своє відео доступним для перегляду в Інтернеті, тому він опублікував свій вміст на YouTube. Пізніше він використовував програму для малювання під назвою SmoothDraw, і тепер використовує такт Wacom, щоб малювати за допомогою ArtRage. Навчальні посібники записуються на комп'ютер.

Позитивні відгуки спонукали хана закінчити свою роботу в 2009 році та зосередити увагу на навчальних посібниках (які потім вийшли під назвою Khan Academy) на повний робочий день. Хан Лаб Школа, школа, заснована Сал Хан і пов'язана з Khan Academy, відкрилася 15 вересня 2014 року в Маунтін В'ю, Каліфорнія. У червні 2017 р. Ханська академія розпочала відеоролик фінансової грамотності для випускників коледжу, шукачів роботи та молодих фахівців.

1.8 Ресурс EdX для онлайн навчання

EdX – майданчик відкритої освіти, що передбачає безкоштовні онлайн-курси кращих університетів світу (рис.1.8). Доступний також у вигляді додатків для Android та iOS. EdX – це масовий провайдер відкритого онлайн-курсу (МООС). Він проводить онлайн-курси вищих навчальних закладів у широкому колі навчальних дисциплін для студентів у всьому світі, включаючи курси безплатно. Він також проводить дослідження в галузі навчання на основі того, як люди використовують свою платформу. EdX є неприбутковою організацією та працює на вільній програмній платформі Open open source openX [8].

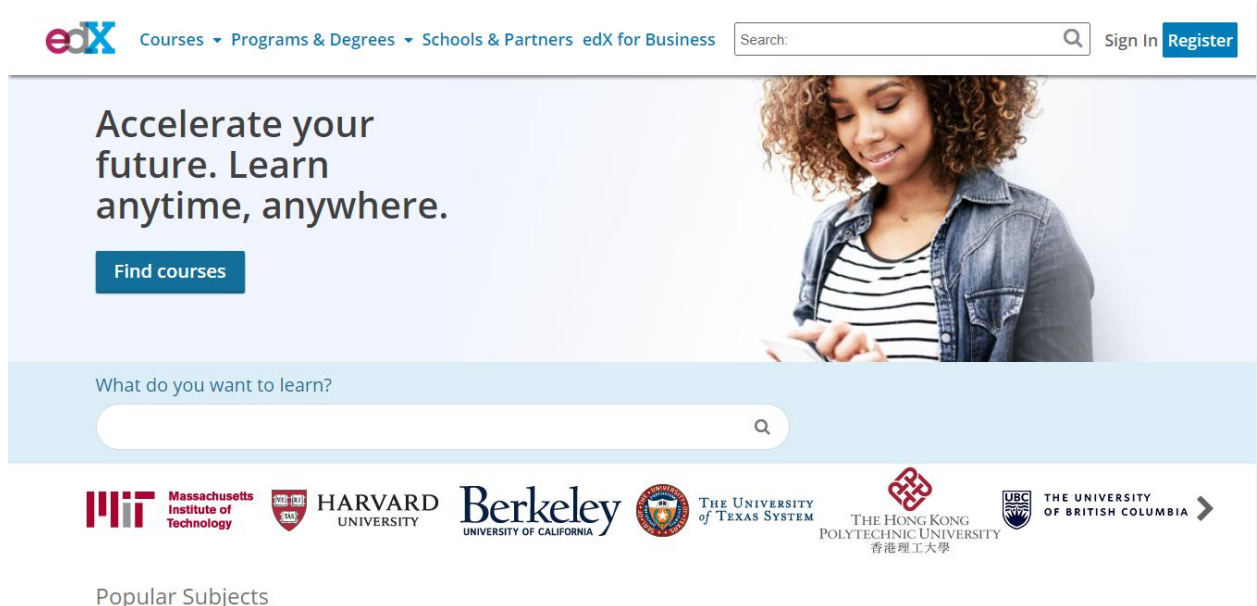


Рисунок 1.8 – Головна сторінка ресурсу EdX

У травні 2012 року в Массачусетському технологічному інституті та Гарвардському університеті створено EDX. Більше 70 шкіл, некомерційних організацій та корпорацій пропонують або планують пропонувати курси на веб-сайті edX. Станом на 29 грудня 2017 року в EdX налічується близько 14 мільйонів студентів, які відвідують понад 1800 курсів в Інтернеті.

Курси EdX складаються з щотижневих навчальних послідовностей. Кожна послідовність навчання складається з коротких відеороликів, що містяться в інтерактивному навчанні, де студенти можуть негайно практикувати концепції з відео. Курси часто включають навчальні відеоролики, схожі на невеликі дискусійні групи на навчальному закладі, онлайн-підручник та форум для обговорення в Інтернеті, де студенти можуть публікувати та переглядати питання та коментарі один одному та навчальні асистенти. У разі необхідності, онлайн-лабораторії включені в курс. Наприклад, в першому MOOC EdX – схемах та курсах електроніки – студенти створили віртуальні схеми в онлайн-лабораторії.

EdX пропонує сертифікати про успішне завершення, а деякі курси мають право на отримання кредиту. Незалежно від того, чи коледж чи університет пропонує кредити на он-лайн курс, це виключно на розсуд школи. EdX пропонує безліч способів проходження курсів, включаючи перевірені курси, в яких студенти мають можливість перевірити курс (без витрат) або працювати на сертифікованому сертифікаті edX (вартість залежить від курсу). Для курсів, оприлюднених до 7 грудня 2015 р., існує можливість взяти курси кодів на

честь роботи над сертифікатом кодексу честі (без витрат). EdX також пропонує сертифікати XSeries для завершення комплекта з двох до семи верифікованих курсів по одному предмету (вартість залежить від курсів).

1.9 Ресурс CodeCombat для онлайн навчання

CodeCombat – це одна з найпопулярніших ігор, яка навчає програмуванню (рис.1.9). Додаток часто використовується в освітніх закладах для навчання розробці. Містить курси для будь-якого віку. CodeCombat є початковою освітньою ігровою компанією в Сан-Франциско, штат Каліфорнія. Вона створює відео-гру на основі браузера, яка виховує гравців навичками комп'ютерного програмування [9].



Рисунок 1.9 – Головна сторінка ресурсу CodeCombat

CodeCombat був заснований у лютому 2013 року Джорджем Сейнсом, Скоттом Еріксоном та Ніком Зим, який раніше розробляв програму для вивчення мови Skritter. У 2014 році компанія отримала підтримку від початкового етапу венчурного капіталу фірми Y Combinator. За даними CodeCombat, її продукти безкоштовні, і компанія планує заробляти гроші за рахунок набору бонусів, звернувшись до досвідчених гравців до програмних компаній. Тим не менш, CodeCombat ввела щомісячну платну підписку, яка надає доступ до додаткового ігрового вмісту.

Гра CodeCombat – це браузерна рольова гра, яка викладає мови програмування Javascript і Python, а також основи інформатики. Щоб просунути рівень гри, гравці повинні довести свої знання, написавши код. Вона включає як одиночні, так і багатокористувацькі компоненти, і орієнтована на аудиторію середньої школи. Гра була позитивно переглянута PC Magazine, хоча рецензент зазначив, що вміст гри був трохи схожий на "загальну випадкову браузерну гру" і "в основному незабутнім".

У січні 2014 року CodeCombat зробило своє програмне забезпечення відкритим вихідним кодом і випустив редактора рівнів, щоб користувачі могли створювати власні ігрові вміст.

1.10 Ресурс CodeWars для онлайн навчання

CodeWars – це сервіс, на якому можна змагатися у програмуванні з іншими учасниками. Користувачам доступні різні мови для написання коду і завдання, які необхідно виконати (рис.1.10).

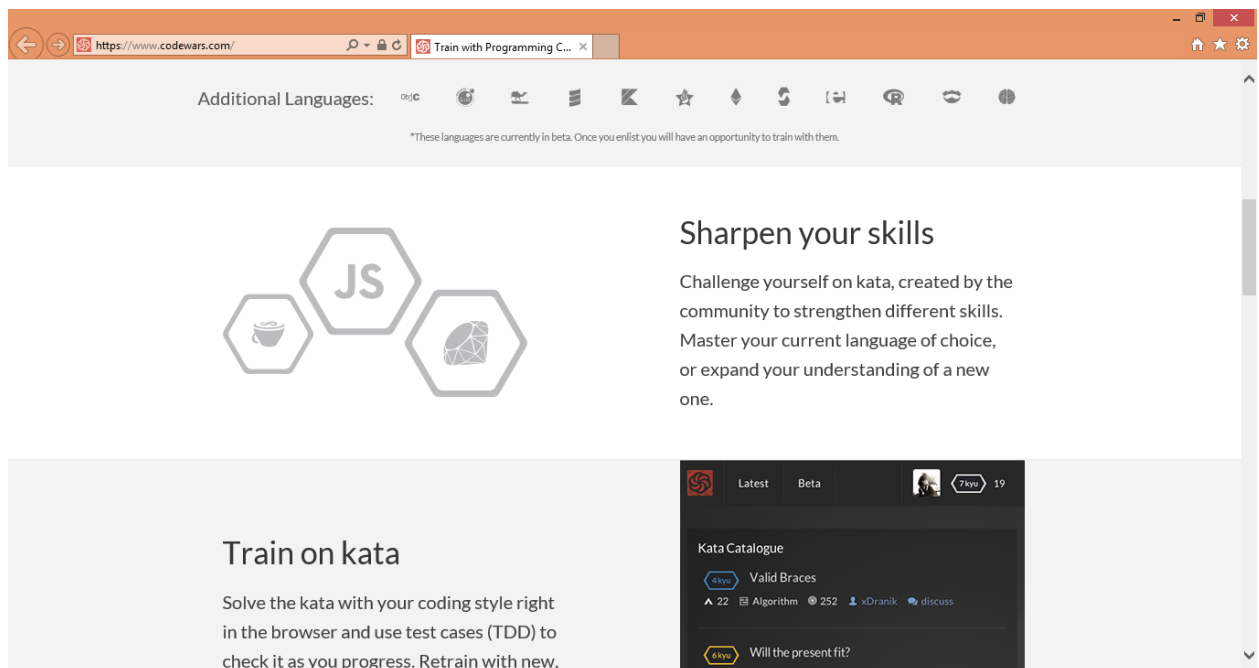


Рисунок 1.10 – Головна сторінка ресурсу Coursera

Codewars – освітня спільнота для комп'ютерного програмування [10]. На платформі розробники програмного забезпечення готуються до програмних завдань, відомих як ката. Ці дискретні програмування навчають різноманітні навички на різних мовах програмування та виконуються в рамках онлайн IDE.

Компанія Codewars була заснована Джейком Хоффнером та Натаном Доктором у листопаді 2012 року. Компанія виробила більше 1 мільйона доларів у фінансуванні ряду інвесторів у технологічному просторі, включаючи Cornerstone OnDemand Inc., венчурний фонд Venture51, Пайг Крейг та Брайан Лі.

1.11 Порівняльний аналіз функціоналу систем для вивчення мов програмування

В рамках проведеного дослідження різних систем для навчання, складемо таблицю з найважливішим функціоналом для будь-якої інформаційної системи навчання мов програмування.

Результати порівняльного аналізу функціоналу сучасних пошукових систем по зображеннях наведені в табл. 1.1.

Таблиця 1.1 – Порівняльний аналіз функціоналу сучасних пошукових систем

Система	Udacity	Ln	Javv	Tynker	EdX	Khan	Coursera
Сертифікування	+	+	+	+	–	+	–
Тестування	+	+	+	+	–	–	–
Теорія	+	–	–	–	+	–	+
Дизайн	+	+	+	+	+	–	+
Можливість програмування в реальному часі	+	–	–	–	–	+	+
БД	+	+	+	–	–	+	+
Офлайн робота	–	–	–	+	–	–	+
Писочниця	–	+	–	+	–	+	–

Аналіз показав, що більшість існуючих систем здійснюють вивчення та закріплення матеріалу за допомогою веб додатків та їх аналогів мобільних додатків не так і багато.

Але є перні плюси деяких платформ. Так як є можливість проглядати теоретичний матеріал та проходити тестування. Всі створені роботи зберігаються в БД. Є можливість їх як і зберігати так і видаляти, але є мінуси: нема можливості експортувати та імпортувати данні і продукти при навчанні; немає структури програми для більш вдалого вивчання; немає так названої пісочниці

для офлайнового програмування. На сьогоднішній день всі програмні продукти які схожі на запланований мною більш схожі на ігри.

Найбільш вдалим продуктом, в якому є все що дійсно потрібно для легкого розуміння та вивчення необхідної інформації є Lrn. Але в німу немає можливості програмування в реальному часі та добре продуманих різних варіантів завдань і тестів для практики. Але в ньому дуже добре зібрана базова інформація, але вона більш схожа на підручник.

Таким чином, питання створення програмного забезпечення, що дозволяє отримати повноцінні знання з програмування за заданим зразком, залишається відкритим.

2 ОГЛЯД ОСНОВНИХ МЕТОДОЛОГІЧНИХ ПІДХОДІВ НАВЧАННЯ

2.1 Системний підхід у навчанні

Сутність підходу – відносно самостійні компоненти розглядаються як сукупність взаємопов'язаних компонентів. Суб'єкти педагогічного процесу – педагог і учень, зміст освіти, методи, форми, засоби педагогічного процесу. Завдання вихователя – облік взаємозв'язку компонентів.

Системний підхід орієнтує на виділення в педагогічній системі, перш за все інтеграційних інваріантних системоутворюючих зв'язків і відносин; на вивчення і формування того, що в системі є стійким, а що змінним, що головним, а що другорядним. Він передбачає з'ясування внеску окремих компонентів-процесів в розвиток особистості як системного цілого. В цьому відношенні він дуже тісно пов'язаний з особистісним підходом, який означає орієнтацію при конструюванні і здійсненні педагогічного процесу на особистість як мету, суб'єкт, результат і головний критерій його ефективності. Він настійно вимагає визнання унікальності особистості, її інтелектуальної та моральної свободи, права на повагу. Він передбачає опору в освіті на природний процес саморозвитку задатків і творчого потенціалу особистості, створення для цього відповідних умов.

2.2 Особистісний підхід

Сутність: визнає особистість як продукт суспільно-історичного розвитку і носія культури, і не допускає зведення особистості до натури.

Особистість як мета, суб'єкт, результат і головний критерій ефективності педагогічного процесу.

Унікальність особистості – її інтелектуальна моральна свобода, право на повагу. Завдання вихователя: створення умов для саморозвитку задатків і творчого потенціалу особистості.

Особистісний підхід вимагає визнання унікальності особистості, її інтелектуальної та моральної свободи, права на повагу. Він передбачає опору в вихованні на природний процес саморозвитку задатків і творчого потенціалу особистості, створення для цього відповідних умов.

2.3 Діяльнісний підхід у навчанні

Сутність: діяльність – основа, засіб і умова розвитку особистості, це цільне перетворення моделі навколишньої дійсності.

Завдання вихователя – вибір і організація діяльності дитини з позиції суб'єкта пізнання праці і спілкування (активність самого). Це передбачає: усвідомлення, цілепокладання, планування діяльності, її організація, оцінка результатів і самоаналіз (рефлексія).

А.Н. Леонтьєв. «Для оволодіння досягненнями людської культури, – писав він, – кожне нове покоління повинно здійснити діяльність, аналогічну (хоча і не тотожну) тієї, яка стоїть за цими досягненнями». Ось чому, щоб підготувати вихованців до самостійного життя і різнобічної діяльності, необхідно в міру можливостей залучити їх в ці види діяльності, тобто організувати повноцінну в соціальному і моральному відношенні життєдіяльність.

Діяльнісний підхід вимагає перекладу дитини в позицію суб'єкта пізнання, праці і спілкування. Важливим при цьому є те, що активність особистості, її потреби в самовдосконаленні розглядаються неізолювані. Вони розвиваються лише в умовах взаємин з іншими людьми, побудованих за принципом діалогу. Діалогічний підхід в єдності з особистісним і діяльнісним становлять сутність методології гуманістичної педагогіки.

2.4 Індивідуальний підхід у навчанні

Це психолого-педагогічний принцип, в якому постулюється важливість для навчання і виховання обліку індивідуальних особливостей кожної дитини. У педагогічній роботі вихователь (учитель) організовує свою діяльність з урахуванням особливостей кожного вихованця т. К. Все діти різні, а значить і здатності у них теж різні.

Виділені методологічні підходи педагогіки як галузі гуманітарного знання дозволяють:

- 1) визначити її дійсні проблеми і способи їх вирішення;
- 2) проаналізувати всю суму освітніх проблем і встановити їх порядок значущості (ієрархію);
- 3) реалізувати гуманістичну парадигму освіти.

Сприйняття як безпосереднє відображення світу класифікується за різними підставами.

Традиційно виділяють п'ять видів сприйняття відповідно до провідним аналізатором, які беруть участь в побудові перцептивного образу (по модальності сприйняття):

- візуальне;
- аудіальное;
- дотикове (тактильну);
- смакове;
- нюхові.

Розрізняють також види сприйняття в залежності від об'єкта сприйняття, наприклад, сприйняття простору, часу, руху, швидкості; творів живопису, музики; основних явищ соціального життя людини (іншої людини, подій суспільного життя) і т.п.

Візуальний канал. Спрямований на засвоєння інформації шляхом більшого зосередження на зорових образах. У людини, якого переважає даний канал сприйняття, відзначається висока здатність засвоювати інформацію через читання. В даному випадку особистості досить прочитати матеріал, і інформація міцно «закріпиться» в мозку. Немає необхідності переказувати прочитане або ділитися з оточуючими. Якщо інформація сама по собі носить суперечливий характер, викликає додаткові питання, провокує суперечка, то в індивіда може виникнути потреба докладно ознайомитися з різними думками для того, щоб сформулювати свою точку зору.

Аудіальний канал. Спрямований на засвоєння інформації шляхом концентрації переважно на слухових образах. Якщо переважає даний канал сприйняття, у людини відзначається висока здатність до запам'ятовування через прослуховування потрібного матеріалу. Студенти, у яких домінує аудіальний канал, прекрасно засвоюють запропоновану інформацію під час лекції і можуть будинку нічого не вчити – все і так легко вляглося в голову, так що не залишилося зайвих питань! Якщо виникають складні моменти, матеріал складний і незрозумілий, така особистість зазвичай прагне відразу уточнити важливі деталі і розібратися на місці, задавши лектору відповідні питання.

Кинестетический канал. Спрямований на засвоєння інформації шляхом зосередження переважно на фізичних відчуттях. Кинестетическое сприйняття тісно пов'язане з органами дотику, тому такій людині під час розмови обов'язково потрібно торкатися до співрозмовника. Запах, смак теж мають для цієї особистості першорядне значення – вона більш за все уважна до деталей і власним почуттям. Якщо ви запитаете людину, що з ним відбувається, він зможе в фарбах описати свої емоції, розпізнати їх справжні прояви.

Дігітескій канал. Спрямований на засвоєння інформації шляхом концентрації на абстрактно-логічних образах. Така людина схильна у всьому шукати сенс, розкласти свої знання «по полицках». Дігіталу вкрай важливо знати, з якою метою він виконує ту чи іншу дію і що з цього буде. Він має здатність прогнозувати ситуацію, а тому схильний до планування і глибокого аналізу подій, що відбуваються. Найчастіше, дігітали по життю займаються науковою діяльністю.

Сприйняття навколишнього світу, як правило, комплексно; воно являє собою результат спільної діяльності різних органів почуттів. Більш того, сприйняття складних явищ предметного і соціального світу здійснюється насамперед завдяки участі процесів пам'яті, мислення та уяви. Інакше кажучи, вести мову про процес сприйняття в «чистому вигляді» у багатьох випадках неправомірно.

У психології існує розподіл видів сприйняття в залежності від участі в ньому інших психологічних утворень: емоційне сприйняття (сприйняття світу, мистецтва); раціональне сприйняття (сприйняття, підпорядковане процесу мислення) і ін.

Кожен з видів сприйняття має свої специфічні особливості і механізми. Їх опис представляє задачу як психології, так і інших галузей знання: фізіології, кібернетики.

В моїй роботі я задію два види сприйняття візуальне та тектильне. Вони допоможуть великій кількості людей сприймати та зачувати інформацію, а візуальний простий інтерфейс полігшує цей процес.

2.4.1 Візуальне сприйняття інформації

Людський мозок обробляє дані паралельно з візуальним сприйняттям, різні ділянки мозку одночасно активуються через мережі нейронів, і тому реакція мозку дуже швидка.

Зорове сприйняття – це вулиця з двостороннім рухом. З одного боку, ми бачимо найдрібніші деталі навколишнього середовища і тут же інтерпретуємо їх в єдине ціле. З іншого, ми звертаємося до нашої пам'яті, тобто в ту частину мозку, де зібрані всі патерни нашого знання світу, і в залежності від сьогохвилинних цілей інтепретіруем побачені дані.

Сприйняття інформації людиною – це об'єднання висхідних і низхідних процесів мозку.

Людина зберігає інформацію в різних розділах пам'яті. Сенсорна (короткострокова) пам'ять записує швидкоплинні враження останніх мілісекунд. Вона дозволяє згадати що-небудь, «записане» не пізніше 1 хвилини. Існує гіпотеза, що короткочасна пам'ять спирається більшою мірою на акустичний (вербальний) код для зберігання інформації і в меншій мірі – на зоровий.

Оперативна пам'ять це робочий простір, в якому ми аналізуємо, синтезуємо інформацію і управляємо їй. Ця пам'ять допомагає нам зрозуміти світ, порівнюючи те, що ми бачимо, з тим, що вже знаємо, – об'єднання спадного і висхідного процесів обробки інформації.

При надходженні нової інформації, наш мозок ідентифікує її з уже зберігається в пам'яті. Якщо збіги перебувають, мозок ідентифікує об'єкти і образи, доповнюючи таким чином існуючі знання. Якщо збігів немає, мозок робить відповідні висновки про нову інформацію.

Все це відбувається в оперативній пам'яті дуже швидко, нова інформація або доповнює щось вже відоме, або піддається подальшій обробці, так і залишаючись в оперативній пам'яті. Саме тому нам доводиться повторювати номер телефону кілька разів, щоб його запам'ятати.

За осмислення інформації відповідають психологічні моделі, які формуються у людини за роки пізнання світу. Наприклад, у нас є модель того, як працює веб-сайт: це навігаційне меню і посилання, і ця модель допомагає нам користуватися різними сайтами в різних місцях. Але це відбувається тільки тому, що всі сайти влаштовані за принципом меню-посилання. Психологічні моделі легко передаються від одного об'єкта до іншого, поки ці об'єкти використовують один і той же шаблон виконання.

Щоб обробка інформації робочою пам'яттю пройдена, т. е. відповідності знайдені, така інформація передається в довгострокову пам'ять. І коли надходить нова інформація, мозок знову кодує її – шукає відповідності в довгостроковій пам'яті.

Виходить, чим більше знайомих і зрозумілих людині асоціацій міститься в рекламному повідомленні, тим вірніше, що він запам'ятає це повідомлення. Саме тому аналогії і метафори так добре працюють, адже чим більше нова інформація буде пов'язана з раніше накопченими знаннями, тим більше ймовірно збереження мозком цієї інформації. Пам'ятаєте? "Повторення мати навчання". Саме постійні повторення інформації різними способами сприяють тому, що інформація буде передана в довгострокову пам'ять.

Приклад: 7-значний номер може бути запомнен в короткочасної пам'яті і забутий через кілька секунд. З іншого боку, людина може пам'ятати його за рахунок повторення протягом тривалого часу.

І схеми і моделі мислення у кожної людини свої, але вони схожі у людей, об'єднаних якимось загальним ознакою - читають один і той же журнал або відвідують один і той же бійцівський клуб. Тобто правильно підібрані об'єкти, форми, кольору та ін. будуть активізувати схеми і психологічні моделі цільової аудиторії. Якщо зрозуміти особливості мислення, когнітивні особливості аудиторії, тоді відповідні асоціації в рекламному повідомленні зроблять його особливо ефективним. Власне, тут і бере початок класифікація цільової аудиторії за характерними ознаками.

Рівень розвитку – класифікувати аудиторію за ступенем просунутості і зробити дизайн у відповідності з цим.

Відволікання – менше відволікаючих елементів – більше фокусування. Не створювати дизайн, який збиває з пантелику.

Візуальна грамотність – зрозумітийте, наскільки зрозумілі і знайомі будуть ті чи інші символи аудиторії.

Мотивація – переваги аудиторії допоможуть визначити, як її мотивувати. Велика мотивація призводить до більшої уваги, і як наслідок – до глибшого розуміння рекламного повідомлення.

Культура, традиції – це певні кольори. Різні кольори інтерпретуються по різному в різних культурах.

2.4.2 Дотикове (тактильне) сприйняття інформації

Тактильні відчуття — це відчуття доторку і тиску. У взаємодії людини з довкіллям вони відіграють велику роль, оскільки сигналізують про присутність того чи іншого подразника, який контактує з поверхнею тіла.

Дотикові (тактильні) відчуття виникають унаслідок дії механічних подразників на поверхню шкіри. Вони належать до групи шкірних відчуттів. Необхідною умовою їхнього виникнення є контакт з об'єктивним подразником, тиск на шкіру, що спричинює її деформацію. Периферичний тактильний аналізатор створюють нервові закінчення в шкірі. Останні з'єднуються з мозковою частиною аналізатора за допомогою провідного шляху (нервові чутливі нервові волокна), який проходить через спинний мозок, стовбур головного мозку, зоровий бугор у кору великого мозку. Відсутність відчуттів тиску й ваги називають абарестезією, м'язово-суглобового відчуття – абатестезією.

Значна кількість інформації від оператора через візуальні та слухові канали викликає серйозні перевантаження. Крім того, через наслідки певних перешкод, сигнал цієї форми може сильно спотворюватися та викликати помилки при розпізнаванні інформації. У зв'язку з цим останнім часом відбувається пошук можливості передачі інформації про інші канали розпізнавання інформації оператором. Найбільш перспективним є використання тактильного аналізатора.

Експериментальні дослідження показали, що дотичні зображення формуються на основі синтезу значної кількості тактильних та м'язових сигналів. Відомо, що шкіра людини визнає термічні, хімічні, механічні та електричні подразники. Якщо перші два способи передачі інформації ще не доступні, є певні досягнення щодо останніх двох (47 100).

Механічні подразники передаються з допомогою вібратора і виявляються різними способами різними частинами шкіри тіла. Абсолютна чутливість вимірюється як мінімальний тиск, необхідний для відчуття. Найчутливіша область (губи, язик) – 1 ... 50 мг / мм², найбільш чутлива область (спина, верхня) – 10 г / мм².

Диференціальний диференціальний поріг становить близько 7% від початкового тиску.

Космічна чутливість також залежить не тільки від характеристик подразника, але і від характеристик конкретної області людського тіла. Диференціальний пороговий простір становить принаймні від 1 до 2,5 мм від губ і пальців, а максимум – 60 мм від спини та плечей. Найвища чутливість спостерігається при частоті вібрації 100 ... 300 Гц.

Відомий цікавий спосіб передачі інформації через вібратор. Людська мова записується у фільмі і відтворюється повільніше, ніж звичайний темп. Отриманий низькочастотний електричний сигнал змінюється у механічну вібрацію пластини, торкаючись поверхні шкіри людини. Після деякого тренування оператор може визначити стандартний звук мови. Цей метод може використовуватися для передачі сигналів у середовищах із значним шумом, коли аналізатор слуху є неефективним.

У випадку людей із вадами зору (слабозори та зору) роль тактильного аналізатора широко використовується, оскільки інформація про навколишнє середовище є єдиним каналом, який передається в мозок людини.

Характеристики тактильно-вібраційної чутливості людини розглядаються при побудові оригінального пристрою для людей з вадами зору "Optoson", в

яких оптичний сигнал перетворюється в тактильну вібрацію, надаючи сліпим можливість прочитати існуючу книгу. Швидкість – 40-50 слів за хвилину.

Кожне досягнення в цьому напрямку має деякі недоліки, які обмежують використання механічних методів передачі інформації. По-перше, це неповнота самого вібратора, тобто об'єм та інерція. У зв'язку з цим розвивається використання стимулів електричного стимулювання для передачі інформації з імпульсами прямокутного струму. Залежно від розміру імпульсу існує три порогови відчуття:

- абсолютні порогови, при яких людина відчуває дію стимуляторів;
- порог болю, коли людина відчуває неприємну відчуття;
- поріг болю, який не можна переносити, коли людина припиняє діяти на стимулятори.

Використання електричного стимулятора вимагає попередньої підготовки, після чого абсолютний порог зменшується, а інший поріг підвищується, але тоді час майже не змінюється і не залежить від області стимулу. Старі люди вищі, ніж молодші.

Важливою перевагою електричного подразнення шкіри в порівнянні з механічною шкірою є потреба в силі сигналу (100 разів) та можливість використання дрібних електродів. Це дуже важливо при розробці пристроїв для передачі інформації через тактильний аналізатор. У Кореї та за кордоном розроблені тактильні стимулятори та тактильні кодекси, які покращують доставку інформації водіям.

Різні канали передачі інформації використовують різні аналізатори, але вони не функціонують окремо, але вони працюють в одній дуже складній системі. У цьому випадку вплив подразника на конкретний аналізатор впливає не тільки на пряму відповідь, але і на функцію всіх інших аналізаторів. Наприклад, чутливість центрального поля залежить від впливу гучного звуку, а збільшення запаху, солодкого смаку, комфортного положення тіла, атмосферного тиску або подразнення шкіри зменшує чутливість навколишнього поля.

2.5 Тестування у навчальному процесі

Тест – це теоретично і практично обґрунтована система висловлювань, завдань, що дозволяє отримати вимірювання відповідних психологічних властивостей.

Відомо, що термін «тест» походить від англійського слова «test» - випробування, проба. Історія зберегла ім'я людини, який першим використову-

вав слово «тест» в якості наукового терміна, що означає психолого-діагностичну методику. Це заслуга американського психолога, першого в світі професора психології Дж.М.Кеттелла (1860-1944). У 1890 році в одному з наукових журналів їм була надрукована стаття «Mental Tests and Measurements». Вчений розробив серію тестів (50) і використовував їх у створеній ним лабораторії (1891) при Колумбійському університеті для оцінки рівня інтелектуального розвитку.

У шкільній практиці тести використовувалися набагато раніше. У ряді джерел можна знайти згадку про те, що тести були вперше застосовані в 1862 році у Великобританії. З метою перевірки знань учнів учитель Грінвічської школи Дж. Фішер створив градуированную книгу, яка містить питання-завдання і відповіді на них (учні повинні були вибрати правильну відповідь). Книга Фішера охоплювала такі предмети як граматики, математика, французьку мову, святе письмо, загальна історія, малювання та ін..

Теорія тестування розвивалася в роботах зарубіжних психологів ХХ століття А. Біне, В.Штерна, Ч.Спирмен і ін. Основоположником тестології вважається англійський психолог Ф. Гальтон (1822-1911), який вніс істотний внесок в розробку теорії тестування, тестових завдань і їх застосування для педагогічних цілей.

В даний час під тестами розуміються стандартизовані завдання, призначені для перевірки знань, умінь, професійної підготовки в усіх областях.

Існує кілька підходів до класифікації тестів. По предмету дослідження виділяють три групи тестів:

1) Общелічного тести. З їх допомогою вивчають психічні властивості особистості як цілісне утворення.

2) Особистісні тести. Призначені для діагностики тієї чи іншої риси, характеристики, властивості (інтелектуальних і спеціальних здібностей, рівня загальної відповідальності, самоконтролю, властивостей особистості). Особливо широке застосування ця група тестів знайшла при вимірюванні психічних процесів (пам'ять, увага, мислення та ін.)

3) Групові тести. Застосовуються при діагностиці групових психічних процесів - рівня згуртованості груп і колективів, особливостей групового психологічного клімату, міжособистісних взаємин, сили «тиску» групи на неї членів та ін.

2.5.1 Методи генерації тестових завдань

Метою завдання є аналіз існуючих методів тестування та розробка моделі, яка буде являти собою автоматизований тест з наступними вимогами:

- легкість створення бази знань, що буде основою створення тестових робіт;
- можливість налаштування різних типів завдань тестування;
- ясність словника;
- гнучкі можливості тестування на основі важливих сфер навчального вмісту, цілей та стратегій тестування;
- можливість використання результатів тестування для адаптації індивідуальних навчальних процесів;
- модель може бути інтегрована з мережевою ієрархічною мережевою моделлю навчальних матеріалів.

Щоб удосконалити та спростити процес підготовки дистанційних курсів за допомогою тест-блоків, вчені запропонували різні підходи до автоматизації створення тестових завдань. Існують методи перевірки параметрів, які відносно легко реалізувати з вашою аудиторією. Ключ до цього методу – забезпечити шаблон для декількох студентів, які можуть бути різними для певних параметрів, які автоматично генеруються. Відповідь вводиться на клавіатурі. Тому, згідно з певною формулою або алгоритмом, система замінює параметри, так що кожен учень отримує мету особи та студент може отримати правильну відповідь, щоб підтвердити відповідь. Недоліком цього підходу є вузьке націлювання. Тому параметри моніторингу та тестування підходять для програми точних наукових і практичних методів для тестування знань з теоретичних знань та функцій управління.

Підхід до використання семантичних мереж для тестування автоматизації був отриманий з вивчення автоматизації управління знаннями. Основна семантична мережа акордів. Є кілька ключових моментів так званого 1 природного небінарного "процесу" і "програми". У цьому випадку ви можете встановити зв'язок між типами "часткового типу". Тоді ви розумієте, що суть "процедури" є "частиною" сутності "програми". Тестова робота полягає в тому, щоб пропустити одну з тріадних посилань і запитати, чи немає посилань. Перевага такого підходу полягає у спроможності системи виходити з знань області. Недоліком є висока вартість розробки повної семантичної мережі, яка точно відображає предметну область дослідження. Іншим недоліком такого підходу є неоднозначність мови, а іноді і неможливість формованої роботи. Таким чином, на основі семантичної мережі часто виникає питання про об'єкт функціональної області, який не має вивчення індивіда в контексті освітнього

значення. Цей недолік виникає з проблем, які характеризують класичну модель знання штучного інтелекту, яку можна назвати проблемою всіх знань. Оскільки дані опитування відбуваються у певних ситуаціях, поведінковий тест для поширених запитань, таких як це сталося, через вимоги до мережі, успішність студентів та недоліки навчання, значення повного відображення формату домену не відповідає. Поінформованість чи комплексне формулювання вимог потребують значних зусиль для створення правильної моделі без належної підготовленості у навчальних цілях та тестуванні.

Ці та інші питання притаманні здатності застосовувати класичну модель II до навчальних завдань. Незважаючи на те, що дві моделі можуть забезпечити дводоменну систему вирішення, перша мета другого випадку, симулювання "знаю", а другий – мета моделювання, але знання штучного інтелекту та освіти. Система моделювання дає можливість "вчити" людські знання про певну область, яка може мати суттєві відмінності тут. Суттєвою відмінністю у створенні цього завдання є необхідність розробки спеціалізованих моделей формалізації знань для навчання та організації на основі автоматизованого тестування. Основним засобом доставки наукової інформації студентам в дистанційну освіту є форма тексту, яку можна зрозуміти у світлі цього навчання. З огляду на це, доцільно зосередитись на форматі моделі, яка автоматизує конфігурацію тестових операцій на основі тексту.

Формалізація уроку тексту. Тезисова концептуальна модель (ПТМ) розроблена на перетині багатьох наукових дисциплін, включаючи наступні тексти: знання напрямку та інженерії штучного інтелекту; Освіта, тобто уроки, що розкривають правила навчання - тлумачення (тлумачення), вивчення правил інтерпретації текстів; Лінгвістика, її семантика, регулярність природної мови, зміст лексичної єдності, сенс і тлумачення. Інструменти впровадження - це технології, що розробляють веб-системи.

У програмі можна виділити чотири компоненти, такі як зв'язок між учасниками програми, передачу навчальної інформації, такої як управління класом, знання та соціальні компоненти. Перша місія дистанційної освіти полягає у спрощенні та оптимізації поширення навчальної інформації, знань студенту. Історично склалося так, що найбільш рання форма передачі інформації в традиційних формах навчання є усною: наприклад, читачі читають лекції. Натомість основним форматом передачі дистанційної навчальної інформації від віддаленого вчителя є текстовий або, швидше, гіпертекстовий та мультимедійний вміст. Іншими словами, основним засобом передачі знань, або "прото-

колом", є текст, що містить навчальну інформацію. Це є основою концептуальної моделі знань тезауруса.

Основою структури ПТМ є така ж сама, як і в деяких областях знання концепцій, дискусійних тем та навчальних матеріалів. Ця концепція обговорюється і вказує на певний предмет і надається студентам. Наприклад, в курсі «Алгоритмізація та програмування» можна виділити такі поняття, як «процедура», «цикл», «програма», «змінна», «життєвий цикл програми». У процесі "Програмування в середовищі Java" можна виділити такі поняття як "об'єкт", "подія", "клас", "форма", "компонент JButton". Набір системних понять відображається наступним чином:

$$C = (C_1, \dots, C_{n1})$$

Включає резюме, спеціальні елементи структури для подання знань концепцій ПТМ. Цей документ є концептуальним твердженням. Цесу можна порівняти з символами, атрибутами чи твердженнями про цю концепцію. Це поняття відноситься лише до тем, що розглядаються в тексті, але цей документ є знаннями або технічним. Повнота цих наборів пропозицій залежить від повноти бази знань та здатності системи освіти генерувати ефективні тестові роботи. Фактично цей білий документ містить одне або декілька речень, які безпосередньо описують конкретну концепцію, але саме поняття не відображається у словнику. З точки зору лінгвістики ця стаття в принципі виключена. Ось кілька прикладів: Документ про поняття "процедури" – "Програми можна розділити на підпрограми"; "Клас" – "Тези про поняття методів, тобто функції та процедури", а також поля-атрибути можуть бути включені в структуру. Підсумок резюме системи виглядає наступним чином.

$$T = (T_1, \dots, T_{n2})$$

Поняття та резюме називаються РТ елементами. Кожна стаття пов'язана з концепцією. Це з'єднання забезпечується наступним співвідношенням:

$$ST: T \rightarrow C$$

У свою чергу, кожне поняття може мати будь-яку кількість абстракцій, описаних відносинами:

$$TC: C \rightarrow 2^T$$

Як зазначалося вище, основним вектором знань є навчальний матеріал. Таким чином, вибір конкретної семантичної одиниці тісно пов'язаний з процесом маніпулювання підручниками. Підготувавши систематичні інгредієнти вуличного курсу, всі інгредієнти в підсумку поділяються на дрібні шматки, також називають пострілами. Тому на кожному етапі навчальної інформації студенти отримують невелику частину, яка дозволяє їм краще зосередити увагу та визнати поданий матеріал. Концепція рамки дуже важлива в ПТМ. Це забезпечує зв'язок між значущими елементами та масою. У цій статті ми не обмежуємось конкретною реалізацією загальної структури збереження та подання тексту. ПТМ також дозволяє іншим реалізаціям цієї структури інтегрувати ПТМ з існуючими системами дистанційного навчання. ПТМ є одним з компонентів системи освіти з іншими моделями. Включає конструювання навчальних матеріалів з використанням спеціальних моделей вмісту. Позначимо частину чи сторінку навчального вмісту.

$$V = \{V_1, \dots, V_{n3}\}$$

Семантичні елементи ПТМ призначаються безпосередньо з тексту фрагмента навчання. Сам процес формування концептуального тезауруса насправді читає текст значущим чином і вимагає простих маніпуляцій. Тому при вивченні підручника вчитель призначає атрибути безпосередньо до тексту та додає важливу концепцію навчання та статтю до БЗ. В результаті кожний фрагмент V_i може бути джерелом певної кількості первинних T_j .

$$TV: V \rightarrow 2^T$$

Кожен T_j складається з короткого шматку V_i :

$$VT: T \rightarrow V$$

Оскільки дисертація належить до єдиного навчального поля, поняття та поняття, що застосовуються для багатьох освітніх полів, і зв'язки між підручниками, надаються опосередковано через підручники з концептуальних дисциплін. Поняття, пов'язані з цією сферою освіти, визначаються оператором.

$$CV(v) = \{c: TV(v) \cap TC(c) \neq 0\}$$

Як результат, навчальний матеріал, що впливає з цієї концепції, визначається оператором.

$$VC(c) = \{v: TV(v) \cap TC(c) \neq 0\}$$

Схематична схема площі ПТМ для тексту показана на рис. 2.1.

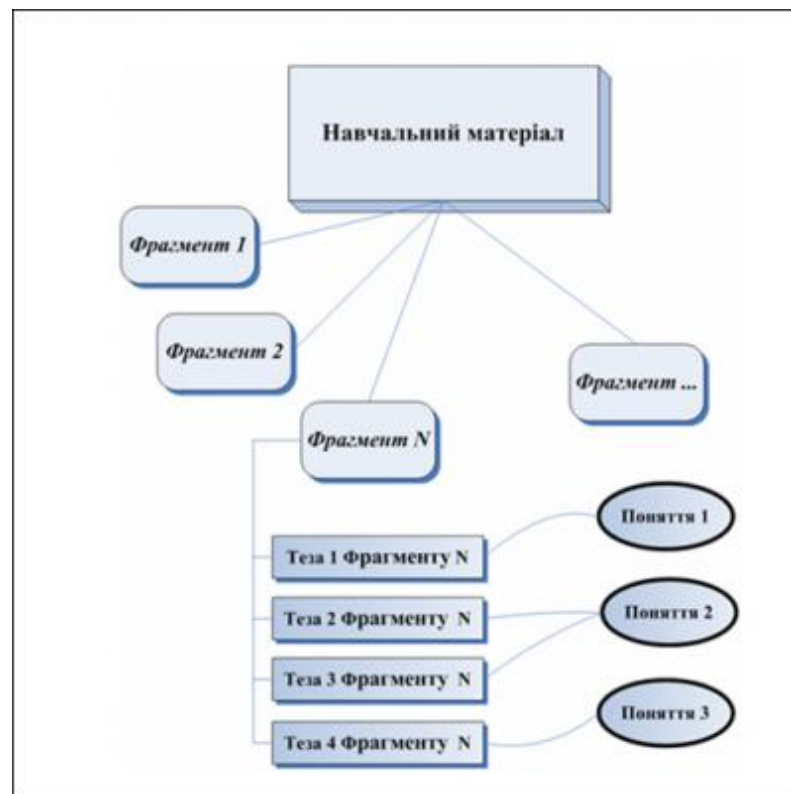


Рисунок 2.1 – Ділянка ПТМ у співвідношенні з навчальним матеріалом

Поняття і тези можуть бути віднесені до певних класів. Ця класифікація служить для збереження в БЗ інформації про смисловий або лексичний характер того чи іншого поняття чи тези. Класи понять і тез використовуються в алгоритмах побудови тестових завдань.

Тезисна концептуальна модель забезпечує вирішення проблем сумісності існуючого штучного інтелекту та інтелектуальних систем навчання шляхом формування знань з викладання тексту, який забезпечує основне джерело знань. Формування ПТ-баз здійснюється шляхом змістовного текстового читання, що значно спрощує процес створення знань.

ПТМ забезпечує дистанційне тестування генератора системи навчання, завдяки унікальності кожного тесту на конфігурацію, що дозволяє звести до мінімуму проблему великих банківських проблем, неможливо несправедливого тестування, статичного тестування або інспекції. PST – це достатньо, щоб розвинути мережевий домен, оскільки він коштує менше, ніж достатньо, щоб створити великі банківські та невеликі витрати на оплату праці одночасно з витратами на оплату праці. На відміну від тестів, сгенерованих в семантичній мережі, лексичний запас на основі ММТ активно впливає на контроль якості. Точна передача семантичних даних у сферу навчальних матеріалів дозволяє проводити різноманітні стратегії тестування форматної оцінки. Таким чином, ПТМ дозволяє створювати тести в певних областях навчального вмісту, а також у результатах тестування, ви можете точно сприймати освітні концепції, які потребують оновлення студентів та тренування фрагментації. Ці можливості можуть бути використані для подальшого розвитку індивідуальних навчальних процесів на основі аналізу результатів тестування.

ПТМ інтегрований з ієрархічною мережевою моделлю дерева вмісту навчання - Інтернетом для створення персоналізованих навчальних середовищ та функцій, таких як створення окремих тестів та створення окремих курсів.

Фактична реалізація підходу ПТМ та автоматизоване тестування виконуються в рамках Порталу дистанційного навчання. Інструменти впровадження програмного забезпечення Включає асинхронний веб-інтерфейс та модуль створення і аналіз тестів на базі даних РТ-додатків для ММТ.

Більше досліджень має бути спрямовано на підвищення якості тестової роботи та вдосконалення методів та алгоритмів для цього покоління. Що є перспективним – це розширити базову класифікацію тез і концепцій та класифікувати нові шаблони для тестування. Іншим перспективним напрямком є створення моделі шаблонів для відкритого тестування.

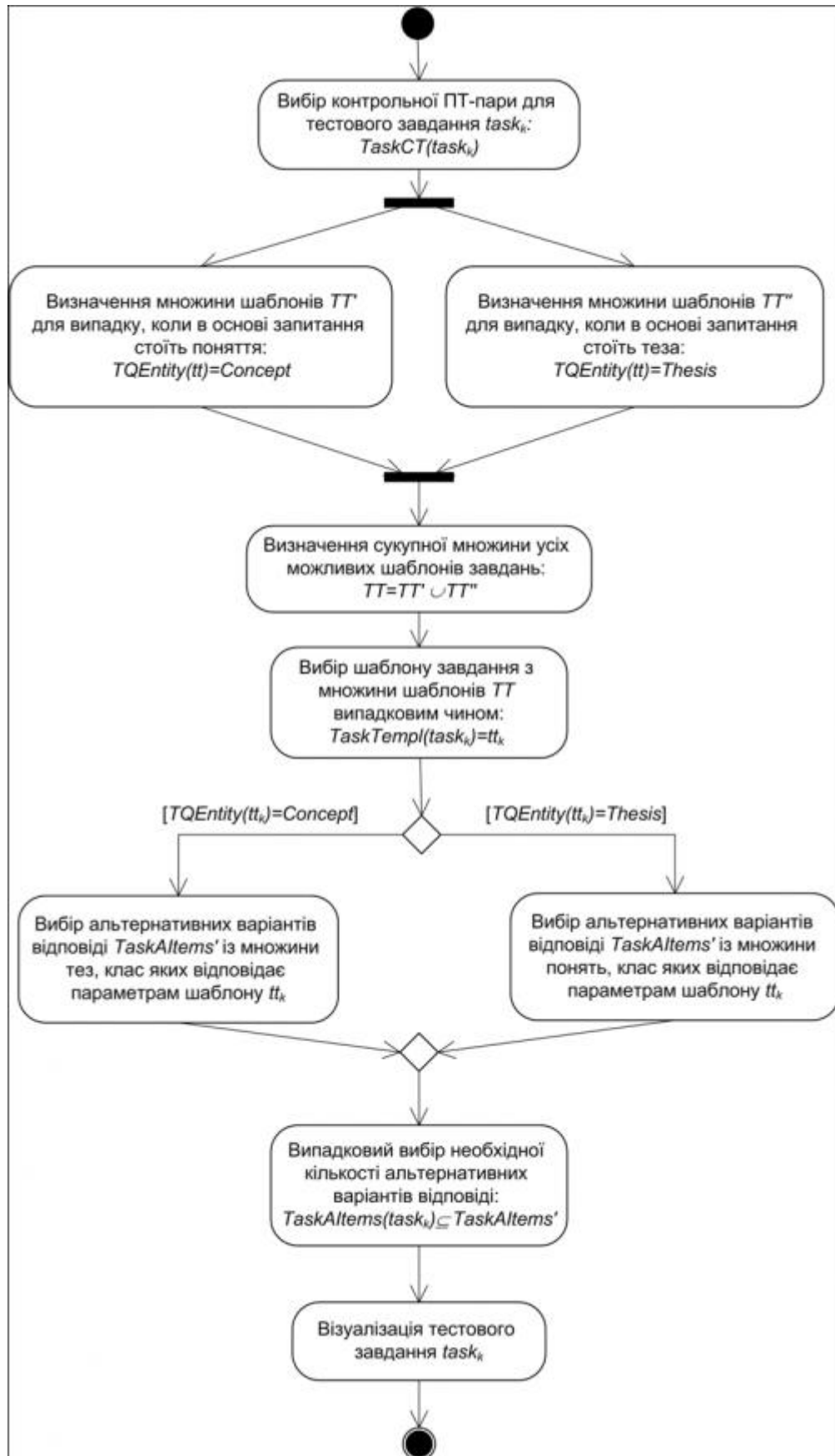


Рисунок 2.2 – Генерація тестового завдання. Діаграма діяльності у нотації UML

2.5.2 Drag-and-drop технологія

Drag-and-drop (в перекладі з англійської означає буквально тягни-і-кидай; Бери-і-Кинь) – спосіб оперування елементами інтерфейсу в інтерфейсах користувача (як графічним, так і текстовим, де елементи GUI реалізовані за допомогою псевдографіки) за допомогою маніпулятора «миша» або сенсорного екрану.

Спосіб реалізується шляхом «захоплення» (потримавши головною (першою, частіше лівою) кнопкою миші) відображуваного на екрані комп'ютера об'єкта, програмно доступного для подібної операції, і переміщенні його в інше місце (для зміни розташування) або «кидання» його на інший елемент (для виклику відповідного, передбаченого програмою, дії). По відношенню до вікон (також здатним до переміщення подібним способом) цей термін зазвичай не вживається.

Базовими діями і найпростішими прикладами drag-and-drop дій є: переміщення об'єкта, переміщення об'єкта з панелі в панель, хоча в сучасних операційних системах drag-and-drop набув широкого застосування і є одним з головних способів взаємодії з комп'ютером в графічному інтерфейсі користувача (рис.2.3).

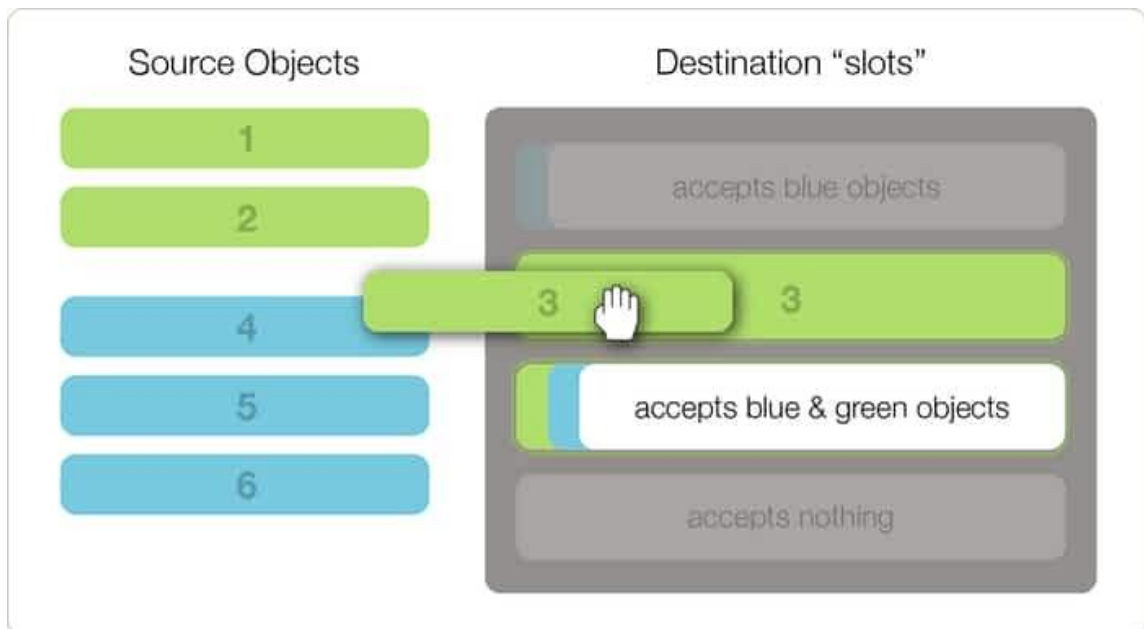


Рисунок 2.3 – Принцип роботи Drag-and-drop технології

Об'єктами для переміщення можуть бути наступні елементи інтерфейсу: значки (іконки) Робочого столу, плаваючі панелі інструментів, ярлики програм в панелі завдань (починаючи з Win XP), елементи TreeView, текстовий рядок, комірка DataGridView., Також елементи OLE. Переміщатися об'єкти можуть як в межах деякої певної області, в межах одного вікна, між панелями одного вікна, так і між різними вікнами.

Основи Drag і Drop використання функціональності drag і drop має на увазі виконання наступних кроків:

Визначити стерпний елемент. Присвоїти true атрибуту draggable елемента, який ми хочемо перенести. Для детальної інформації дивіться The Draggable Attribute.

Визначити дані, які можуть бути переміщені, вони можуть бути різного формату. Наприклад, текстові дані, що містять рядок тексту який може бути переміщений. Для детальної інформації дивіться Drag Data.

Визначити зображення яке буде поруч з покажчиком миші протягом всієї операції перетягування. Якщо для користувача зображення не буде визначено, буде згенеровано картинка за замовчуванням, в залежності від елемента, на якому була затиснута кнопка миші (що означатиме, що елемент переносять). Ознайомитися детально з установкою зображення перетягування можна за посиланням Setting the Drag Feedback Image.

Визначити можливі ефекти переносу. Можливі три таких ефекту: сору показує, що переміщуються дані копіюються з колишнього місця розташування в нове, move показує, що переміщуються дані повністю переносяться на нове місце, і link показує, що створюється якась форма взаємодії або зв'язку між вихідною точкою і точкою призначення. Протягом операції переміщення, картинка яка слідує за курсором миші може змінюватися в залежності від того, чи може елемент бути переміщений в область під курсором. Якщо перенесення дозволений, переміщення може бути вироблено. Дивіться Drag Effects для детальної інформації.

Визначити область призначення. За замовчуванням браузер не дозволяє переміщати що-небудь на HTML елемент. Однак, щоб зробити елемент активним для переміщення інших елементів на нього, просто скасуйте дію за замовчуванням. Тобто, підпишіться на події "ondragenter" або "ondragover". Для детальної інформації дивіться Specifying Drop Targets.

Обробити завершення перенесення. Ви можете отримати дані з перенесеного елемента і зробити над ними необхідні операції. Для детальної інформації, будь ласка, дивіться Performing a Drop.

Для того, щоб ознайомитися із загальним списком даних підтримуваних операцією drag and drop дивіться Recommended Drag Types.

Також доступні приклади з кращою практикою використання операції drag and drop для переміщення даних різних типів:

- Text;
- Links;
- HTML and XML;
- Files;
- Images;
- Document Nodes.

Ряд подій спрацьовують протягом всієї процедури drag and drop. Drag-події спрацьовують протягом операції переміщення; події миші, такі як mousemove – немає. Також запам'ятайте, що події dragstart і dragend не спрацьовують при спробі перенести файл з операційної системи в браузер.

Властивість dataTransfer всіх подій переміщення містить дані про всі drag і drop операції.

Dragstart спрацьовує коли елемент почав переміщатися. У момент спрацьовування події dragstart користувач починає перетягування елемента. Оброблювач цієї події може бути використаний для збереження інформації про переміщуваний об'єкт, а також для зміни зображення, яке буде асоційоване з переміщенням. Дане подія не спрацьовує, коли деякий файл буде переноситися з операційної системи в браузер. Для детальної інформації Starting a Drag Operation.

Dragenter спрацьовує, коли переміщуваний елемент потрапляє на елемент-призначення. Оброблювач цієї події показує, що елемент знаходиться над об'єктом на який він може бути перенесений. Якщо ж обробника немає, або він не робить ніяких дій переміщення за замовчуванням заборонено. Ця подія також використовується для того, щоб підсвітити або промаркувати об'єкт над яким відбувається переміщення в разі, якщо переміщення на даний елемент дозволено. Для детальної інформації дивіться Specifying Drop Targets.

Dragover – дана подія спрацьовує кожні кілька сотень мілісекунд, коли переміщуваний елемент виявляється над зоною, яка приймає перетягуються елементи. Для детальної інформації дивіться Specifying Drop Targets.

Dragleave – ця подія запускається в момент перетягування, коли курсор миші виходить за межі елемента. Обробникам слід прибрати будь-яку підсвічування або інші індикатори, які вказували на присутність курсора, щоб тим самим позначити реакцію на припинення перетягування.

Drag опускається при переміщенні елемента або виділеного тексту. С цією технологією можливо зробити велику кількість тестів для мобільного додатку:

- перетягування блоків з правильною відповіддю;
- перетягування блоків з правильною відповіддю в потрібному порядку;
- перетягування слів в приготівленний текст на місця з пропусками.

2.6 Sandbox – рантайм компілятор коду

Інтерпретація – порядковий аналіз, обробка та виконання вихідного коду програми або запиту (на відміну від компіляції, де весь текст програми, перед запуском, аналізується і транслюється в машинний або байт-код, без її виконання).

Можна нескінченно дивитися на вогонь, воду і активність програм, ізолюваних в пісочниці. Завдяки віртуалізації ти одним кліком можеш подивитись результат набраного тобою кода.

Втім, віртуалізація застосовується і в дослідницьких цілях: наприклад, захотілося тобі проконтролювати вплив свежескомпілірованої програми на систему або запустити дві різні версії додатка одночасно. Або створити автономне додаток, яке не буде залишати слідів в системі. Варіантів застосування пісочниці - безліч. Чи не програма диктує свої умови в системі, а ти їй вказуєш дорогу і розподіляєш ресурси.

Пісочниця – спеціально виділена середовище для безпечного виконання комп'ютерних програм. Зазвичай являє собою жорстко контрольований набір ресурсів для виконання гостьової програми - наприклад, місце на диску або в пам'яті. Доступ до мережі, можливість спілкуватися з головною операційною системою або зчитувати інформацію з пристроїв введення зазвичай або частково емулюють, або сильно обмежують. Пісочниці представляють собою приклад віртуалізації.

Простий інтерпретатор аналізує і тут же виконує (власне інтерпретація) програму покомандно (або через підрядник), у міру надходження її вихідного коду на вхід інтерпретатора. Перевагою такого підходу є миттєва реакція. Недолік - такий інтерпретатор виявляє помилки в тексті програми тільки при спробі виконання команди (або рядка) з помилкою.

Інтерпретатор компілюючого типу – це система з компілятора, який переводить вихідний код програми в проміжне представлення, наприклад, в байт-код або р-код, і власне інтерпретатора, який виконує отриманий проміж-

ний код (так звана віртуальна машина). Перевагою таких систем є більшу швидкодію виконання програм (за рахунок виносу аналізу вихідного коду в окремий, разовий прохід, і мінімізації цього аналізу в інтерпретаторі). Недоліки – більша вимога до ресурсів і вимога на коректність вихідного коду. Застосовується в таких мовах, як Java, PHP, Tcl, Perl, REXX (зберігається результат парсинга вихідного коду), а також в різних СКБД.

За рівної кількості інтерпретатора компілює типу на компоненти виходять компілятор мови і простий інтерпретатор з мінімізованим аналізом вихідного коду. Причому вихідний код для такого інтерпретатора не обов'язково повинен мати текстовий формат або бути байт-кодом, який розуміє тільки даний інтерпретатор, це може бути машинний код якийсь існуючої апаратної платформи. Наприклад, віртуальні машини на зразок QEMU, Vochs, VMware включають в себе інтерпретатори машинного коду процесорів сімейства x86.

Деякі інтерпретатори (наприклад, для мов Лісп, Scheme, Python, Бейсік і інших) можуть працювати в режимі діалогу або так званого циклу читання-обчислення-друку (англ. Read-eval-print loop, REPL). У такому режимі інтерпретатор зчитує закінчену конструкцію мови (наприклад, s-expression в мові Лісп), виконує її, друкує результати, після чого переходить до очікування введення користувачем наступної конструкції.

В мої роботі я вибрала Jbed (емулятор java).

JBed (Oberon Microsystems) – створювалася на базі BlackBox, підтримує спільне виконання модулів компонентни Паскаля і класів Java, є найбільш якісною реалізацією Java2 Micro Edition, яку використовують 5 з 6 відомих виробників стільникових телефонів.

3 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА РЕЗУЛЬТАТІВ

3.1 Структура розробленого програмного забезпечення

Повсякденне використання модних гаджетів давно увійшло до звички сучасних людей. Багато хто вважає за краще користуватись різними корисними програмками для спілкування окрім звичних дзвінків і смс. Тому розробка мобільних додатків для різних цілей користується значним успіхом. Не має особливого значення.

Розробка мобільних додатків відіграє все більш важливу роль. На сьогоднішній день існує великий вибір мов програмування для розробки мобільних додатків. Це пов'язано з тим, що для різних мобільних. Новітні інформаційно-комунікаційні технології в освіті пристроїв доводиться використовувати різні мови програмування, що обумовлене тим, що мобільні пристрої мають різні операційні системи (ОС). Цільова платформа (або платформи) – iOS, Android, WindowsPhone, BlackBerry – буде мати значний вплив на вибір мови програмування.

Для реалізації програмного забезпечення для навчання була обрана об'єктно-орієнтована мова програмування Java з наступними бібліотеками: Support Library, Android Annotations, MaterialDrawer, Glide. Середа розробки – Android Studio. Засіб розробки інтерфейсу – Android XML Templates. Сбірник проекту – Gradle. З додаванням Sandbox інтерпретатором.

Розроблена система навчання людей програмуванню, імплементовані всі функції для комфортної роботи без постійного знаходження у мережі інтернет, створена для вивчення потрібної мови в будь-якому місці з умовою наявності смартфона на додатку до нього.

Мобільний додаток був розроблений з максимально інтуїтивно розумілим дизайном, та комфортним та простим інтерфейсом хоча і дуже інформативним.

3.1.1 Support – бібліотеки підтримки

При розробці додатків, які підтримують кілька версій API, ви можете стандартний спосіб, щоб забезпечити нові можливості на більш ранніх версіях Android або граціозно падають назад в еквівалентній функціональності. Замість того, щоб будувати код для обробки більш ранніх версій платформи, ви

можете використовувати ці бібліотеки, щоб забезпечити той рівень сумісності. Крім того, бібліотеки підтримки надають додаткові класи зручності і можливостей, які недоступні в стандартному Framework API для спрощення розробки та підтримки через кілька пристроїв [11].

Спочатку один двоична бібліотека для додатків, бібліотека Підтримки Android перетворилася в набір бібліотек для розробки додатків. Багато з цих бібліотек тепер настійно рекомендується, якщо не важливо, частина розробки додатків.

Цей документ містить огляд бібліотеки підтримки, щоб допомогти вам зрозуміти, його компоненти і як ефективно використовувати його в своєму додатку.

Є кілька різних застосувань для бібліотек підтримки. Зворотні класи сумісності для більш ранніх версій платформи є тільки один з них. Ось більш повний список способів, якими можна використовувати бібліотеки підтримки в додатку:

- зворотна сумісність для нових API-інтерфейсів – велика кількість бібліотек підтримки зворотної сумісності для нових каркасних класів і методів. Наприклад, Fragment клас підтримки забезпечує підтримку фрагментів на пристроях, що працюють під управлінням версією раніше, ніж Android 3.0 (рівень API – 11);

- зручність і допоміжні класи – підтримка бібліотеки надають ряд допоміжних класів, в зокрема, для розробки користувальницького інтерфейсу. Наприклад RecyclerView клас надає інтерфейс віджет користувача для відображення і управління дуже довгі списками, корисних на версії Android від рівня API 7 і вище [12];

- налагодження, тестування і утиліти – є цілий ряд класів, які забезпечують корисність понад коду ви включити в ваше додаток, в тому числі підтримки-анотація бібліотеки для поліпшеної коду линта перевірки на входах методу, Бібліотека тестування підтримки, а також бібліотека Multidex для конфігурації і поширення додатків з більш ніж 65536 методами.

Бібліотеки підтримки надають класи і методи, які близько нагадують API, в Android Framework. Виявивши це, можна здивуватися, якщо треба використовувати рамкову версію API або бібліотеку підтримки еквівалента. Ось керівні принципи для того, коли необхідно використовувати класи бібліотеки підтримки замість Framework API:

- сумісність Специфічна особливість – якщо треба, щоб підтримати не-давню особливість платформи на пристроях, а більш ранні версії платформи, використовують еквівалентні класи і методи з бібліотеки підтримки;
- сумісність для бібліотеки особливості – більш складні допоміжні класи бібліотеки можуть залежати від одного або кількох додаткових класів бібліотеки підтримки, тому ви повинні використовувати допоміжні бібліотеки класів для цих залежностей. Наприклад, ViewPager клас підтримки слід використовувати з FragmentPagerAdapter.

3.1.2 Модуль Android Annotations

Простий код з чіткими намірами є найкращим рішенням для досягнення цих цілей. Щоб полегшити написання і підтримку Android додатків з можливістю використання Java анотації, розробники можуть показати свій намір і дозволити AndroidAnnotations генерувати код автоматично під час компіляції [13].

Можливості:

- впровадження залежностей: ін'єкційні види, додаткові послуги, системні служби, ресурси, тощо;
- спрощена потокова модель: існують анотування, щоб методи виконувались в потоці призначеного для користувача інтерфейсу або в фоновому потоці;
- зв'язування подій: анотовані методи можуть одразу виконувати подію, тобто не потрібно створювати анонімні класи слухачі-подій;
- REST клієнт: створює клієнтський інтерфейс, AndroidAnnotations генерує реалізацію, котру можна реалізовувати для отримання та відправки даних на сервер [14];
- AndroidAnnotations генерує підкласи під час компіляції, тобто можна перевірити код, щоб побачити, як працюють сгенеровані класи.

3.1.3 Система компіляції та збірки проекту Gradle

Gradle – система автоматичного складання, побудована на принципах Apache Ant і Apache Maven, але надає DSL на мові Groovy замість традиційної XML-подібної форми подання конфігурації проекту [15].

На відміну від Apache Maven, заснованого на концепції життєвого циклу проекту, і Apache Ant, в якому порядок виконання завдань (targets) визнача-

ється відносинами залежності (depends-on), Gradle використовує спрямований ациклічний граф для визначення порядку виконання завдань.

Gradle був розроблений для розширюваних багатопроектної збірки, і підтримує інкрементальні збірки, визначаючи, які компоненти дерева збірки не змінилися і які завдання, залежні від цих частин, не вимагають перезапуску.

Основні плагіни призначені для розробки і розгортання Java, Groovy і Scala додатків, але готуються плагіни і для інших мов програмування. Також може бути застосований з фреймворком для тестування додатків JUnit [16].

Під час компіляції додатку може бути використан ProGuard, щоб захистити код від попиток дисасемблювати додаток, ProGuard обфусцює код та не дасть легший шанс добратися до коду зловмиснику [17].

3.1.4 Модуль бокового меню MaterialDrawer

Модуль надає просту створення навігаційної панелі збоку екрану. Існує заголовок з профілями (AccountHeader), MiniDrawer для планшетів (наприклад, як у додатку Gmail), забезпечують призначені для користувача DrawerItems, можливість модифікувати та створювати теми [18].

Можливості:

- найпростіша можлива інтеграція;
- інтегрувати менш ніж за 5 хвилин;
- сумісний з Android API до 14 версії;
- включає в себе AccountSwitcher;
- швидкий і простий API;
- слідує рекомендаціям з проектування Google Material;
- можливість використовувати векторні (SVG) іконки та шрифти за допомогою інтеграції Android-Iconics;
- поставляється з різними темами;
- використовує бібліотеку підтримки AppCompat;
- на основі RecyclerView;
- підтримка RTL;
- можливість використовувати MiniDrawer як у додатку Gmail від Google;
- протестован автоматизованими тестами.

3.1.5 Модуль підтримки векторних іконочних шрифтів Android-Iconics

Ця бібліотека дозволяє включати векторні іконки всюди у проєкті. Ніяких обмежень не наводяться. Бібліотеку можна розширяти своїми векторними шрифтами [19]. У базовій комплектації шрифти не йдуть, тобто їх потрібно підключити вручну.

Можливості:

- немає обмежень щодо розміру, кольору, контуру, фону, відступів, позиціонування;
- одне джерело для іконок (не потрібно більше генерувати MDPI, LXDPI, та інші пакети ресурсів, також зберігати консистентність);
- розширяємість;
- якщо потрібно бовова панель, він також буде працювати з `IconicsDrawable`;
- зберегти в розмірі APK;
- всі ліцензії включені, найкраще використовувати з `AboutLibraries`.

Доступні для використання шрифти:

- Google Material Design Icons;
- Material Design Iconic;
- Fontawesome;
- Meteocons;
- Octicons;
- Material Community;
- Weather Icons;
- Typeicons;
- Entypo;
- Devicon;
- IconFoundation;
- Ionicons.

3.1.6 Модуль завантаження та обробки зображень Glide

Glide є швидкою і ефективною з відкритим вихідним кодом управління медіа та завантаження зображень рамкою для Android, яка обертає медіа декодування, пам'ять і кешування диска, і ресурс пулів в простий і легкий у використанні інтерфейс [20].

Glide підтримує вибірку, декодування і відображення відео кадрів, зображення і анімовані GIF-файли. Glide включає в себе гнучкий API, який дозволяє розробникам підключити практично до будь-якої мережевий стек.

За замовчуванням Glide використовує стек на основі призначених для користувача HttpURLConnection, але також включає в себе комунальні бібліотеки підключити до проекту Volley Google або бібліотеці OkHttp Square замість цього.

Основна увага Glide є на створення прокрутки будь-яких список зображень, як гладко і швидко, наскільки це можливо, але Glide також ефективна практично для будь-якого випадку, коли вам потрібно отримати, змінювати розмір, і відображати віддалене зображення.

Сумісність:

- Android SDK: Glide вимагає мінімального рівня API 10;
- OkHttp 2.x: є необов'язкові залежності доступні звані OkHttp-інтеграції;
- OkHttp 3.x: є необов'язкові залежності доступні так звані OkHttp3-інтеграції;
- Volley: є додаткові залежності для підтримки реактивного програмування;
- круглі картинки: CircleImageView/CircularImageView/RoundedImageView відомо, мають проблеми з TransitionDrawable (.crossFade () з .thumbnail () або .placeholder()) і анімованих GIF-файлів, використовуйте BitmapTransformation (.circleCrop() буде доступний в v4) або .dontAnimate(), щоб виправити цю проблему;
- величезні зображення (карти, комікси): Glide може завантажувати величезні зображення, Субдіскретувати їх, але не підтримує масштабування і панорамування ImageViews, оскільки вони вимагають спеціальної оптимізації ресурсів (наприклад, фрагментація) працювати без OutOfMemoryErrors.

3.1.7 Особливості Android Studio

Android Studio, заснована на програмному забезпеченні IntelliJ IDEA від компанії JetBrains, офіційне засіб розробки Android додатків. Дане середовище розробки доступна для Windows, OS X і Linux. 17 травня 2017 на щорічній конференції Google I/O, Google анонсував мову Kotlin використовуваний в Android Studio офіційною мовою програмування для платформи Android на додаток до Java і C ++ [21].

Нові функції з'являються з кожною новою версією Android Studio. На даний момент доступні наступні функції:

- розширений редактор макетів: WYSIWYG, здатність працювати з UI компонентами за допомогою Drag-and-Drop, функція попереднього перегляду макета на декількох конфігураціях екрану.
- збірка додатків, заснована на Gradle;
- різні види збірок і генерація кількох .apk файлів рефакторинг коду;
- статичний аналізатор коду (Lint), що дозволяє знаходити проблеми продуктивності, несумісності версій і інше;
- вбудований ProGuard і утиліта для підписування додатків;
- шаблони основних макетів і компонентів Android;
- підтримка розробки додатків для Android Wear і Android TV;
- вбудована підтримка Google Cloud Platform, яка включає в себе інтеграцію з сервісами Google Cloud Messaging і App Engine.

Android Studio 2.1 підтримує Android N Preview SDK, а це значить, що розробники зможуть почати роботу зі створення програми для нової програмної платформи.

Нова версія Android Studio 2.1 здатна працювати з оновленим компілятором Jack, а також отримала покращену підтримку Java 8 і вдосконалену функцію Instant Run.

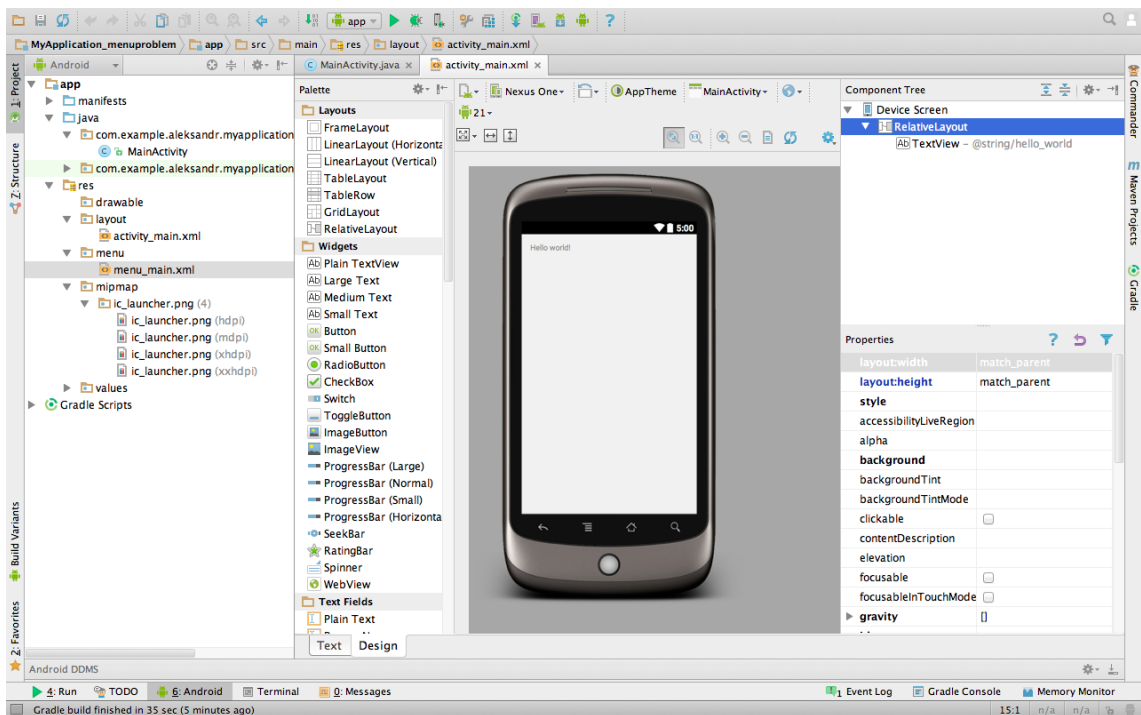


Рисунок 3.1 – Середовище розробки Android Studio

Platform-tools 23.1.0 для Linux без оголошення стала виключно 64-розрядної, навіть при спробі встановити 32-розрядну версію. Іншими словами Android Studio більше не працює (видає непереборні помилки) в 32-розрядних версіях Linux.

В Android Studio 3.0 будуть по стандарту включені інструменти мови Kotlin засновані на JetBrains IDE.

Android – операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux. Підтримується альянсом Open Handset Alliance (ОНА).

Хоча Android базується на ядрі Linux, він стоїть дещо осторонь Linux-спільноти та Linux-інфраструктури. Базовим елементом цієї операційної системи є реалізація Dalvik віртуальної машини Java, і все програмне забезпечення і застосування спираються на цю реалізацію Java.

3.2 Концептуальне проектування бази даних інформаційної системи

База даних є найважливішим елементом будьякої інформаційної системи. Обрана система управління контентом дозволяє плагінам виробляти всілякі операції з базою даних. Для цього в УП2 є свій БД фреймворк ActiveRecord. Сервером бази даних для веб-додатку обрана система керування базами даних – MySQL.

Почнемо з етапу розробки концептуальної бази даних інформаційно-пошукової системи для пошуку вакансій для людей з обмеженими можливостями. Цей етап включає в себе аналіз об'єктів реального світу, які необхідно змоделювати в базі даних і складається з етапів:

- ідентифікація функціональної діяльності предметної області. В даному випадку мова йде про діяльність організації і в якості функціональної діяльності можна розглядати ведення реєстру вакансій в компаніях, котрі мають робочі місця для людей з обмеженими можливостями;

- ідентифікацію об'єктів, які здійснюють цю функціональну діяльність і ідентифікувати всі сутності і взаємозв'язку між ними. Процес "ведення реєстру вакансій" ідентифікує такі сутності: ЗАВДАННЯ, ТЕСТ, КОРИСТУВАЧ;

- ідентифікацію характеристик цих сутностей;

- ідентифікацію взаємозв'язків між сутностями.

Перерахуємо сутності:

– сутність ЗАВДАННЯ може включати такі характеристики як ідентифікатор, назва, тип завдання), повний опис завдання, дата початку роботи;

– сутність ТЕСТ може включати такі характеристики як ідентифікатор, назва позиції, опис завдання, дата проходження, дата модифікації, мінімальна оцінка, оцінка;

– сутність КОРИСТУВАЧ може включати такі характеристики як ідентифікатор, хеш пароля, логін, авторизаційний хеш, електронну пошту, статус, коли створений користувач і коли змінювався;

Наступний етап проектування БД полягає у встановленні відповідності між сутностями і характеристиками предметної області і відносинами і атрибутами в нотації обраної СКБД. Оскільки кожна сутність реального світу володіє якимись характеристиками, в сукупності утворюють повну картину її прояви, можна поставити їм у відповідність набір відносин (таблиць) і їх атрибутів (полів).

Перерахувавши всі відносини і їх атрибути, вже на цьому етапі можна почати усувати зайві позиції. Кожен атрибут повинен з'являтися тільки один раз і треба вирішити, яке відношення буде власником якого набору атрибутів.

На наступному етапі визначаються атрибути, які унікальним чином ідентифікують кожен об'єкт. Це необхідно для того, щоб система могла отримати будь-який одиничний рядок таблиці.

Наступний етап передбачає вироблення правил, які будуть встановлювати і підтримувати цілісність даних. Будучи певними, такі правила в клієнт-серверних СКБД підтримуються автоматично – сервером баз даних; в локальних же СКБД їх підтримку доводиться покладати на користувальницький додаток.

Далі встановлюються зв'язки між об'єктами (таблицями і стовпцями) і виробляється дуже важлива операція для виключення надмірності даних – нормалізація таблиць.

Кожен з різних типів зв'язків повинен бути змодельований в базі даних. Існує кілька типів зв'язків: зв'язок "один-до-одного", зв'язок "один-до-багатьох", зв'язок "багато-до-багатьох".

Зв'язок "один-до-одного" представляє собою найпростіший вид зв'язку даних, коли первинний ключ таблиці є в той же час зовнішнім ключем, що посилаються на первинний ключ іншої таблиці.

Зв'язок "один-до-багатьох" в більшості випадків відображає реальну взаємозв'язок сутностей в предметної області. Вона реалізується вже описа-

ною парою "зовнішній ключ-первинний ключ", тобто коли визначено зовнішній ключ, що посилається на первинний ключ іншої таблиці.

Зв'язок "багато-до-багатьох" в явному вигляді в реляційних базах даних не підтримується. Однак є ряд способів непрямой реалізації такого зв'язку, які з успіхом відшкодовують її відсутність. Один з найбільш поширених способів полягає у введенні додаткової таблиці, рядки якої складаються із зовнішніх ключів, що посилаються на первинні ключі двох таблиць.

Після визначення таблиць, полів, індексів і зв'язків між таблицями слід нормалізувати проєктовану базу даних.

Важливість нормалізації полягає в тому, що вона дозволяє розбити великі відносини, як правило, містять велику надмірність інформації, на більш дрібні логічні одиниці, що групують тільки дані, об'єднані "по природі". Таким чином, ідея нормалізації полягає в наступному.

Кожна таблиця в реляційній базі даних задовольняє умові, відповідно до якого в позиції на перетині кожного рядка і стовпчика таблиці завжди знаходиться єдине значення, і ніколи не може бути безлічі таких значень.

Вигляд проєктованої бази даних після нормалізації демонструють табл. 3.1 – 3.3.

Структурна діаграма бази даних наведена у додатку А.

Таблиця 3.1 – Структура таблиці "TEST"

Ім'я поля	Тип
ID	INT(11)
NAME	VARCHAR(255)
DESCRIPTION	TEXT
CONTENT	TEXT
RATING	TEXT

Таблиця "TEST" містить в собі інформацію про всі тести які ми використовуємо у мобільному додатку.

Для ведення всієї інформації та для якісної роботи всі тести ми тримаємо в одному місці. Це дозволяє нам контролювати процес для продуктивної дії програми що ми і хотіли. Але деяка інформація може бути використана в інших таблицях які наведені нижче.

Таблиця 3.2 – Структура таблиці “TASK”

Ім'я поля	Тип
ID	INT(11)
NAME	VARCHAR(255)
DESCRIPTION	TEXT
CONTENT	TEXT
RATING	TEXT
EXAMPLE	TEXT

Таблиця "TASK" містить в собі інформацію про всі завдання на програмування коду.

Таблиця 3.3 – Структура таблиці “CUSTOMER”

Ім'я поля	Тип
ID	INT(11)
DESCRIPTION	TEXT
EXPERIENCE	DECIMAL(10,2)
ACTIVE	TINYINT(1)
DATE_CREATED	TIMESTAMP
DATE_MODIFIED	TIMESTAMP
USER_ID	TEXT

Таблиця "CUSTOMER" містить інформацію про користувачів.

Заключним етапом є вирішення питань надійності даних і, при необхідності, збереження секретності інформації. Для цього необхідно відповісти на наступні питання:

- хто буде мати права (і які) на використання бази даних;
- хто буде мати права на модифікацію, вставку і видалення даних;
- чи потрібно робити різницю в правах доступу;
- яким чином забезпечити загальний режим захисту інформації.

3.3 Розробка додатку

Мобільний додаток розробляється в сукупності з безліччю тонкощів. Вирішення поставленого завдання в реалізації мобільного додатку необхідно організувати зберігання і обробку інформації, а також нестандартне виведення написаного та інтерпретовано кода на монітор смартфона і здійснення маніпуляцій над ним.

По перше для зручності розділимо нашу задачу на наступні підзадачі:

- реалізація архітектури додатка;
- реалізація функціоналу для зберігання інформації;
- реалізація функціоналу для тестування;
- реалізація функціонала для інтерпретації кода за допомогою Sandbox.

При створенні мобільного додатку окрема увага приділялося його оптимізації. Швидкість завантаження даних завжди була актуальною темою. Існують різні способи зменшення часу – оптимізація параметрів сервера і файлів.

Одним із важливих фактором є швидкість, з якою відповідь формуються на сервері. В рамках етапу оптимізації додатка була проведена оптимізація бази даних і організовано кешування даних.

Дизайн додатку дуже сучасний та простий у використанні наведений на рис 3.2.

Також наводиться загальний алгоритм роботи графічного комплексу в додаткуА.

Частина додатку, що реалізує функціонал роботи з тестовими завданнями наведений на рис. 3.3.

Частина додатку, що показує дизайн та функціонал роботи з теоритичною інформацією наведений на рис. 3.4.

Для найкращого візуально сприйняття Drag&Drop технології наведений на рис 3.5 один з тестів. Та компіляція і запуск користацького коду на елефоні з операційної системи Android на рис 3.6.



Рисунок 3.2 – Перший розділ з теоретичними та тестовими завданнями

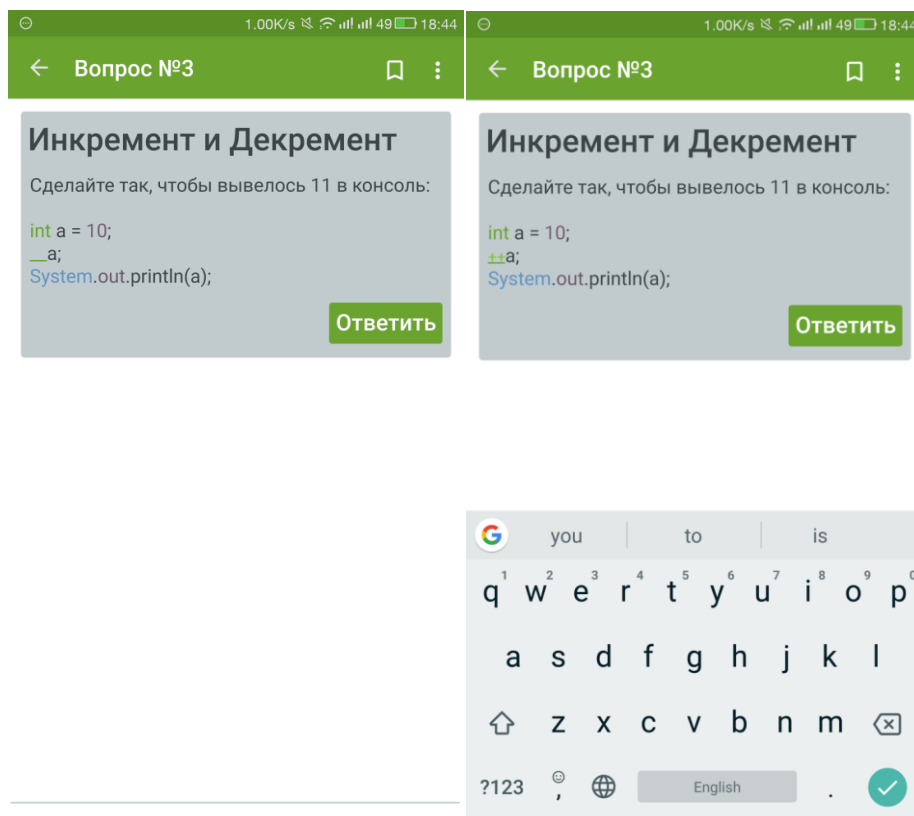


Рисунок 3.3 – Приклад із тестових завдань

1.00K/s 49 18:44

← Раздел №1

Введение в Java

Операторы **инкремента** или **декремента** дают возможность более компактной записи кода для **увеличения** или **уменьшения** значения переменной на один.

К примеру $x=x+1$; может быть спрощено к $++x$;
Пример:

```
int test = 5;
++test; // test is now 6
```

Попробовать

Оператор **декремента** ($--$) используется чтобы уменьшить значение переменной на один.

```
int test = 5;
--test; // test is now 4
```

Попробовать

Далее

Рисунок 3.4 – Представления теоретического материалу

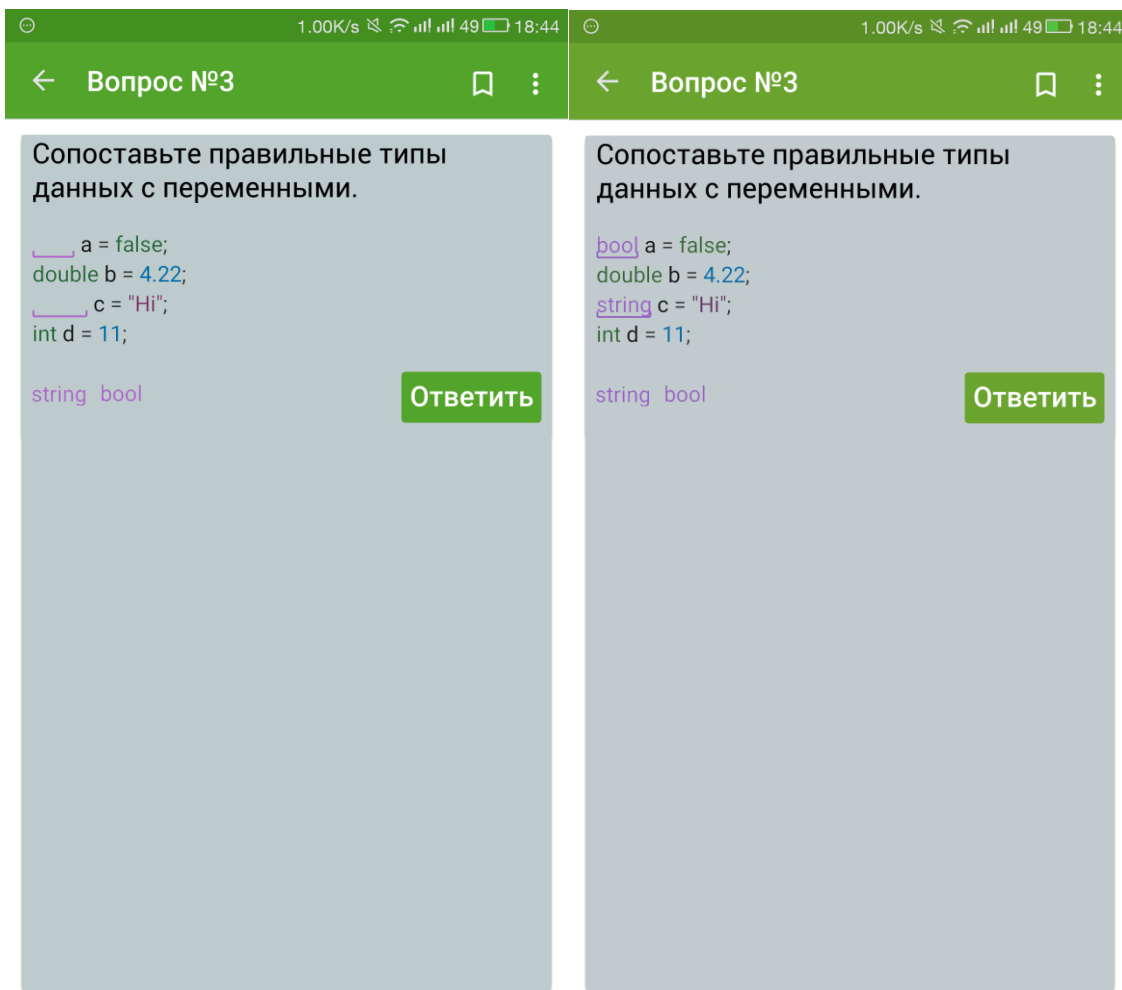


Рисунок 3.5 – Тестування за допомогою Drag&Drop технології

The screenshot shows an Android IDE interface. At the top, there is a green header bar with a back arrow, the text "Пісочниця" (Sandbox), and a three-dot menu icon. Below the header, the code editor displays the following Java code:

```

1 import java.util.Scanner;
2
3 class MyClass {
4     public static void main(String[] args) {
5         Scanner myVar = new Scanner(System.in);
6         System.out.println(myVar.nextLine());
7     }
8 }

```

Below the code editor, there is an "OUTPUT" section containing the text "test string". At the bottom of the screen, there is a toolbar with icons for "ТАВ" (Tab), "{", "}", ";", "=", "(", and a "Запуск" (Run) button with a right-pointing arrow.

Рисунок 3.6 – Компіляція і запуск користувацького коду на телефоні з ОС Android

Більша частина коду була написана за допомогою анотацій котрі додають AndroidAnnotations.

У верхній частині системи розміщується шапка (header), яка містить назву розділу та підрозділу в якому ми знаходимся. Контент поділений на певні частини по тематиці.

Основний макет розбитий на кілька сегментів: шапка сайту (Header), контент (Content). В умовах дизайну буде застосовано багато динамічних об'єктів також буде застосована певним чином анімація.

Розглянемо більш детально. Шапка розбита на декілька блоків: блок з назвою сторінки та кнопками перемикачами зеленого кольору.

Другий блок це основна частина контент.

Далі нашої задачею є первинна розмітка і верстка всіх окремих об'єктів. Які розділи по певним смисловим блокам.

Перше що нам після шапки борсається в очі це блок «Теми з навчанням», який реалізований у вигляді блока що скролиться і дозволяє організувати структуру відповідної інформації.

При відкритті панелі навігації показується меню навігації, в якому 4 розділа. У першому розділі є такі пункти:

- “Основні поняття”;
- “Ведення в Java”;
- “Перша програма”;
- “Компоненти Java”;
- “Переменні”;
- “Приметивні оператори”;
- “Інкремент та декремент”;
- “Строки”;
- “Виключення”.

К кожному Розділу показано кількість теоретичних частин у ньому.

Після нажимання на пункт меню (в звичайному режимі) або подвійного нажимання по екрану (в режимі обмеженого перегляду) відкривається той чи інший розділ, розділ є фрагментом (особливість реалізації для зменшення використаної оперативної пам'яті).

Наприклад при відкритті розділу “Інкремент та декремент”; ми бачимо терію, яку необхідно засвоїти для того щоб зробити завдання.

Завдання тестувальні мають різні види та рівні складнощів. Для повторення і закріплення потрібної нам інформації. Система зроблена таким чином що з самого початку відкрито перше завдання і по проходженню по теорії та проходженню завдання відкриваються завдання подалі. В деякому сенсі це як квест ігра та в кінці ми отримуємо грамоту о проходженні мого курсу по вивченню мови Java.

При реалізації даного додатку я хотіла, щоб додаток був максимально легкий у використанні та швидким для людей. Велика кількість різноманітних

технологій завантаження та обробки зображень зараз існує, але проаналізував весь ринок я знайшла найоптимальнішу.

В моєму випадку Glide технологія є швидкою і ефективною з відкритим вихідним кодом управління медіа та завантаження зображень рамкою для Android, яка обертає медіа декодування, пам'ять і кешування диска, і робить ресурс пулів в простий і легкий у використанні інтерфейс.

Таким чином, в магістерській роботі було розроблено програмну реалізацію програмного продукту, яке дозволяє виконувати його в офлайн в будь-якому місці для вивчення мови програмування. Має великий обсяг теоритичної базової інформації и тести для її закріплення. Підходить додаток і для людей які вперше знайомляться з мовою програмування для засвоєння базових знань та для людей які знають інші мови програмування та бажають дізнатися тонкощі мови програмування Java. Сьогодні весь світ прагне більше знати, встигати та знайти себе в цьому швидкому світі наповненому великими масивами інформації.

ВИСНОВКИ

Для полегшення процесу навчання програмуванню був реалізований мобільний додаток для вивчення мови Java. Легкий, комфортний у використанні та дуже інформативний. В додатку наведені базові конструкції та елементи які присутні в усіх сучасних мовах програмування, що відрізняє його зі всіх схожих додатків. Також є можливість офлайн програмування, що робить його дуже практичним.

У магістерській роботі був виконаний аналіз та програмна реалізація мобільного додатку для вивчення мов програмування з Sandbox.

Також були задіяні бібліотеки: Support Library, Android Annotations, MaterialDrawer, Glide. Середовище розробки – Android Studio. Засіб розробки інтерфейсу – Android XML Templates. Сбірник проекту – Gradle. З додаванням Sandbox інтерпретатором.

Sandbox інтерпретатор це середовище програмування. В якому можливо реалізовувати невеликі задачі з програмування. Вивчивши новий матеріал відразу застосувати

Магістерська робота складається з трьох розділів.

У першому розділі наведений порівняльний аналіз різних відомих систем для вивчення мов програмування. Більшість з них мають схожий функціонал, але вони орієнтовані чи на малих дітей чи на людей вже добре знаючих хоча б базові елементи та правила програмування.

У другому розділі виконаний аналіз потрібної інформації для реалізації додатку. Надані класифікація та робота окремих методів. Визначені їх переваги та недоліки. Надані алгоритми методів Drag-and-drop технології та тлумачення роботи Sandbox.

У третьому розділі роботи виконано проектування програмного забезпечення. Розроблений мобільний додаток. Для розробки програмного забезпечення була реалізована з використанням сучасних програмних засобів. Обрано мову програмування – Java. Середовище розробки – Android Studio. Засіб розробки інтерфейсу – XML. Розроблена програма може працювати з у форматі офлайн.

Ця програма може бути одним з елементів вдосконалення освітнього процесу підготовки молодших спеціалістів з ІТ в коледжах та технікумах. Реалізація зручного інтерфейсу веб-застосування зробить процес навчання більш простим та зрозумілим для початківців програмістів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ресурс Udacity для вивчення програмування [Електронний ресурс] – Режим доступу: <https://www.udacity.com/>
2. Додаток Swift Playgrounds для вивчення мови програмування Swift [Електронний ресурс] – Режим доступу: <https://www.apple.com/swift/playgrounds/>
3. Ресурс Codecademy для вивчення програмування [Електронний ресурс] – Режим доступу: <https://www.codecademy.com/>
4. Додаток Lrn для вивчення мов програмування та гіпер-розмітки [Електронний ресурс] – Режим доступу: <https://itunes.apple.com/us/app/lrn-learn-to-code-in-html-css-javascript-ruby-python/id1019622677?mt=8>
5. Додаток Tynker для вивчення алгоритмізації та програмування [Електронний ресурс] – Режим доступу: <https://www.tynker.com/>
6. Ресурс Coursera для онлайн навчання [Електронний ресурс] – Режим доступу: <https://ru.coursera.org/>
7. Ресурс Khan Academy для онлайн навчання [Електронний ресурс] – Режим доступу: <https://ru.khanacademy.org/>
8. Ресурс EdX для онлайн навчання [Електронний ресурс] – Режим доступу: <https://www.edx.org/>
9. Ресурс CodeCombat для онлайн навчання [Електронний ресурс] – Режим доступу: <https://codecombat.com/>
10. Ресурс CodeWars для онлайн навчання [Електронний ресурс] – Режим доступу: <https://www.codewars.com/>
11. Документація для розробчика [Електронний ресурс] – Режим доступу: http://habrahabr.ru/Developer_Documentation.
12. Android – Вікіпедія [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/wiki/Android>
13. AndroidAnnotations: Fast Android Development. Easy maintainance. [Електронний ресурс] – Режим доступу: <https://github.com/androidannotations/androidannotations>
14. Do you know what a REST API is? [Електронний ресурс] – Режим доступу: <https://www.sitepoint.com/developers-rest-api/>
15. Gradle – Вікіпедія [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/wiki/Gradle>

16. JUnit – Википедия [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/wiki/JUnit>

17. ProGuard | Official website [Електронний ресурс] – Режим доступу: <https://www.guardsquare.com/en/proguard>

18. Модуль бокового меню MaterialDrawer [Електронний ресурс] – Режим доступу: <https://github.com/mikepenz/MaterialDrawer>

19. Модуль підтримки векторних іконочних шрифтів Android-Iconics [Електронний ресурс] – Режим доступу: <https://github.com/mikepenz/Android-Iconics>

20. Модуль завантаження та обробки зображень Glide [Електронний ресурс] – Режим доступу: <https://github.com/bumptech/glide>

21. Android Studio – Википедия [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Android_Studio