

УДК 373.55.016:[004.4:530.145]

Hardware and software tools for teaching the basics of quantum informatics to lyceums students

Liudmyla V. Lehka^[0000–0001–5768–5475],
Svitlana V. Shokaliuk^[0000–0003–3774–1729]

Kryvyi Rih State Pedagogical University, 54, Gagarin Ave.,
Kryvyi Rih, 50086, Ukraine
{asp-18-lehka, shokalyuk}@kdpu.edu.ua

Abstract. The article defines the criteria for choosing a cloud-oriented platform for mastering the basics of quantum informatics by students of a specialized (high) school: cross-browser; intuitive interface; the possibility of free (free) access; access without registration and simplified registration; the presence of a systematized reference system with examples; support for the development of the environment by the developer; support for working in a personal educational environment; support for working with quantum algorithms in graphical mode; automatic conversion of quantum algorithms from graphic format to program code text; support for the Ukrainian-language locale; availability of a mobile application; responsive design. The possibilities of platforms for implementing quantum algorithms from the following companies are analyzed: Microsoft, QuTech, Amazon Braket, IBM. The choice of the IBM Quantum Cloud-based platform is justified. Work at IBM Quantum Composer and IBM Quantum Lab is described. Information about quantum operations and Gates is presented: their designation in IBM Quantum Composer and IBM Quantum Lab; the gate Matrix; and the purpose of the gate. An example of implementing quantum teleportation in the form of a scheme and program is given.

Keywords: quantum calculations, quantum computer, quantum circuit, quantum algorithm, IBM Quantum Experience, Python, Jupyter Notebook.

1 Вступ

Вивчення питань предметних змістових ліній шкільного курсу інформатики, а саме: «Інформаційні технології створення та опрацювання інформаційних об'єктів», «Моделювання, алгоритмізація та програмування», «Телекомунікаційні технології», — здійснюється за підтримки відповідного апаратно-програмного забезпечення (зокрема, Інтернет-сервісів).

З експериментальним упровадженням у навчання питань квантової інформатики учнів ліцеїв [11] постає проблема доцільного та

педагогічно виваженого добору апаратно-програмних засобів підтримки їх вивчення з урахуванням низки критеріїв.

У навчальних програмах з інформатики для ЗЗСО [12] немає обмежень щодо використання учителем різних видів апаратного та програмного забезпечення, за умови його відповідності вимогам чинного законодавства [2, 3, 10, 17]. Також у програмах не зазначаються універсальні критерії добору програмного забезпечення. Проте існують авторські системи критеріїв їх добору, які необхідно розглянути.

О. І. Яценко й О. С. Яценко поділяють критерії обрання програмного забезпечення на три групи:

1. критерії, пов'язані з можливостями мови програмування:

- підтримка запису математичних виразів у математичній формі;
- використовувана модель обчислень (потік даних/потік керування);
- наявність підтримки алгоритмічних конструкцій;

2. критерії, пов'язані з можливістю використання середовища на початковому етапі вивчення мов програмування:

- простота, сучасність і візуальна привабливість інтерфейсу;
- наявність методичних посібників;
- україномовний інтерфейс;
- вартість (безкоштовна/платна);

3. критерії, пов'язані з технологічними аспектами середовища:

- кросплатформність;
- підтримка популярних робототехнічних конструкторів;
- ліцензія (пропріетарна чи відкрита);
- наявність підтримки та розвитку середовища [22, с. 107].

П. Г. Шевчук важливими характеристиками середовища програмування як засобу навчання вважає поширеність, доступність, особливості інтерфейсу, спосіб реалізації, системні вимоги, методичну підтримку, зручний та зрозумілий інтерфейс [18, с. 31].

В. М. Базурін зазначає, що на вибір середовища програмування для використання у процесі вивчення мови програмування впливають такі умови:

- технічні характеристики комп'ютерів і системні вимоги середовища програмування;
- наявність операційних систем і додаткового програмного забезпечення, необхідного для функціонування програмного середовища;
- функціонал програмного середовища;
- інтерфейс програмного середовища;
- наявність документації на програмне середовище;
- наявність навчально-методичного забезпечення;
- рівень компетентності вчителя інформатики [1, с. 15].

Т. А. Вакалюк окреслює такі характеристики, яким має відповідати хмаро орієнтоване середовище навчання: доступність і мобільність; відкритість; цілісність і безперервність; ефективність; систематичність; послідовність і структурованість; інноваційність; інтеграція з хмаро орієнтованими ресурсами; наочність; функціональність; колективність; забезпечення проєктної діяльності; науковість; надійність; комунікаційність; гнучкість та адаптивність; індивідуалізація; наповненість; зручність; доцільність [20, с. 156].

А. В. Ворожбит виділяє такі критерії для використання веб-орієнтованих технологій для створення змісту навчання:

- вартість розроблення;
- гнучкість використання;
- зворотній зв'язок зі здобувачами освіти;
- зрозумілість подання навчального матеріалу;
- педагогічний контроль знань, мотивація до навчання;
- можливість використання мультимедійного динамічного контенту;
- навчальна діяльність здобувачів освіти;
- співпраця вчителя зі здобувачами освіти, здобувачів освіти між собою [21, с. 29].

2 Результати

Вибір хмаро орієнтованої платформи для опанування основ квантової інформатики у закладах загальної середньої освіти обґрунтований застосуванням таких критеріїв:

- кросбраузерність;
- інтуїтивно зрозумілий інтерфейс;
- можливість вільного (безоплатного) доступу;
- спрощена реєстрація;
- наявність систематизованої довідкової системи з прикладами;
- підтримка розвитку середовища розробником;
- підтримка роботи у персональному освітньому середовищі;
- підтримка роботи з квантовими алгоритмами в різних режимах, зокрема, графічному;
- автоматичне конвертування квантових алгоритмів із графічного формату в текст програмного коду;
- підтримка україномовної локалізації;
- наявність мобільного застосунку;
- адаптивний дизайн.

Нами були проаналізовані можливості платформ для реалізації квантових алгоритмів від компаній IBM, Alibaba, Microsoft, Google, Intel, D-Wave Systems, Quantum Circuits, IonQ, Honeywell, Xanadu и Rigetti.

Через хмарну платформу для квантових обчислень Azure Quantum (Рис. 1) компанія Microsoft дозволяє відвідувачам навчитися використовувати комплект Quantum Development Kit, щоб створювати програми для квантового обладнання [13]. Microsoft власного квантового комп'ютера не має, а надає доступ ко квантового обладнання Honeywell Quantum Solutions, IonQ, 1Qbit.

На платформі Quantum Inspire [16] нідерландська компанія QuTech надає безкоштовний доступ до навчальних матеріалів, до квантового симулятора та до квантових чіпів (Рис. 2).

Хмаро орієнтована платформа квантових обчислень Amazon Braket [15] надає можливість доступу до квантового обладнання компаній D-wave, IonQ та Rigetti (Рис. 3).

Компанія IBM є першою компанією, що з 2016 року надає хмарний доступ до власного квантового обладнання [4]. Наразі, на нашу думку, IBM Quantum [8] пропонує найбільші можливості безкоштовної реалізації квантових алгоритмів як на квантових симуляторах, так і на реальних квантових комп'ютерах (Рис. 4).

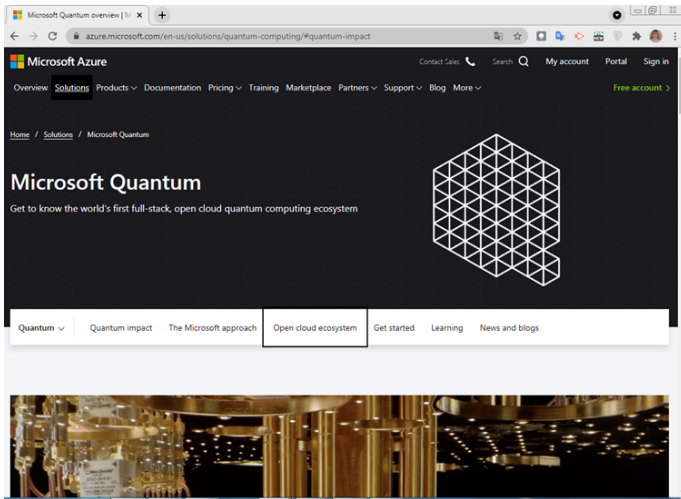


Рис. 1. Головна сторінка хмаро орієнтованої платформи для квантових обчислень Azure Quantum

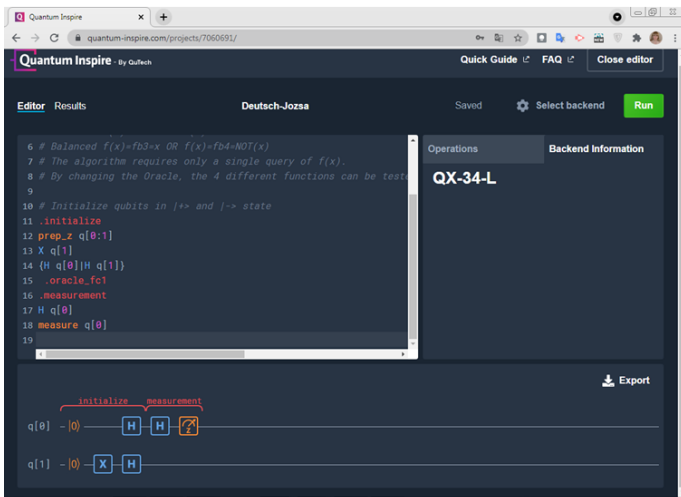


Рис. 2. Фрагмент сторінки хмаро орієнтованої платформи для квантових обчислень Quantum Inspire Quantum

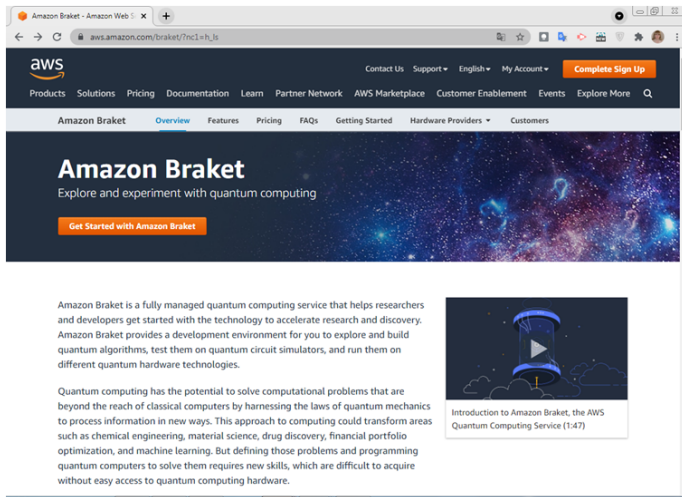


Рис. 3. Головна сторінка хмаро орієнтованої платформи для квантових обчислень Amazon Braket

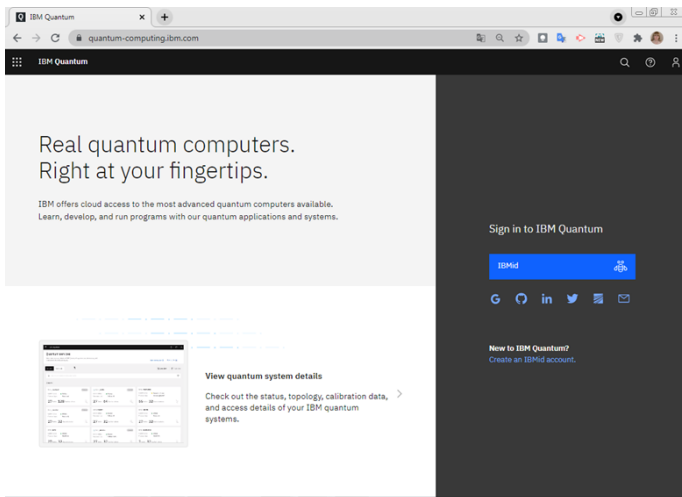


Рис. 4. Головна сторінка хмаро орієнтовної платформи для квантових обчислень IBM Quantum

Узагальнені результати порівняльного аналізу хмаро орієнтованих платформ для роботи з квантовими алгоритмами й організації навчання основ квантової інформатики учнів ліцеїв (за вище означеною авторською системою критеріїв) наведені у таблиці 1.

Таблиця 1. Порівняльний аналіз підходів до забезпечення та платформ для роботи з квантовими алгоритмами

Критерій	Azure Quantum	Quantum Inspire	Amazon Braket	IBM Quantum
Кросбраузерність	+	+	+	+
інтуїтивно зрозумілий інтерфейс	+	+	+	+
спрощена реєстрація	**	+	***	+
можливість вільного (безоплатного) доступу	+	+	+	+
наявність систематизованої довідкової системи з прикладами	+	+	+	+
підтримка розвитку середовища розробником	+	+	+	+
підтримка роботи у персональному освітньому середовищі	*	+	*	+
підтримка роботи з квантовими алгоритмами у графічному режимі	*	-	*	+
автоматичне конвертування квантових схем у програмний код	*	-	*	+
підтримка україномовної локалізації	-	-	-	-
наявність мобільного застосунку	-	-	-	-

Продовження таблиці 1

Критерій	Azure Quantum	Quantum Inspire	Amazon Braket	IBM Quantum
адаптивний дизайн	–	–	–	–

* – інформація потребує уточнення через додаткові умови реєстрації на сервісі

** – ідентифікація за номером телефону або банківською картою

*** – ідентифікація за номером телефону, адресою, банківською картою

Дані таблиці є переконливим свідченням доцільності організувати навчання основ квантової інформатики учнів ліцеїв на платформі IBM Quantum.

Наведемо детальнішу характеристику можливостей інструментарію платформи IBM Quantum.

На час звернення до ресурсу IBM Quantum для дослідження були доступні квантові симулятори від 32 до 5000 кубітів (Таблиця 2). Квантові симулятори працюють швидше, тому спочатку доцільно перевірити свою квантову програму на симуляторі, а вже потім здійснювати звернення до реального квантового комп'ютера.

Таблиця 2. Основні характеристики квантових симуляторів IBM

Назва квантового симулятора IBM	Кількість кубітів	Доступні квантові логічні вентиля
simulator_stabilizer	5000	ID, X, Y, Z, H, S, SDG, SX, SWAP, CX, CY, CZ, DELAY
simulator_mps	100	U1, U2, U3, U, P, CP, CX, CZ, ID, X, Y, Z, H, S, SDG, SX, T, TDG, SWAP, CCX, UNITARY, ROERROR, DELAY
simulator_extended_stabilizer	63	ID, X, Y, Z, H, S, SDG, SX, SWAP, CX, CZ, DELAY, P, CCX, U1, CCZ, T, TDG

Продовження таблиці 2

Назва квантового симулятора IBM	Кількість кубітів	Доступні квантові логічні вентиля
simulator_statevector	32	U1, U2, U3, U, P, R, RX, RY, RZ, ID, X, Y, Z, H, S, SDG, SX, T, TDG, SWAP, CX, CY, CZ, CSX, CP, CU1, CU2, CU3, RXX, RYY, RZZ, RZX, CCX, CSWAP, MCX, MCY, MCZ, MCSX, MCP, MCU1, MCU2, MCU3, MCRX, MCRY, MCRZ, MCR, MCSWAP, UNITARY, DIAGONAL, MULTIPLEX-ER, INITIALIZE, KRAUS, ROERROR, DELAY
ibmq_qasm_simulator	32	U1, U2, U3, U, P, R, RX, RY, RZ, ID, X, Y, Z, H, S, SDG, SX, T, TDG, SWAP, CX, CY, CZ, CSX, CP, CU1, CU2, CU3, RXX, RYY, RZZ, RZX, CCX, CSWAP, MCX, MCY, MCZ, MCSX, MCP, MCU1, MCU2, MCU3, MCRX, MCRY, MCRZ, MCR, MCSWAP, UNITARY, DIAGONAL, MULTIPLEX-ER, INITIALIZE, KRAUS, ROERROR, DELAY

IBM надає відкритий (безкоштовний) доступ до квантових комп'ютерів кількістю від 1 до 32 кубітів (Таблиця 3). Квантові комп'ютери IBM з більшою кількістю кубітів доступні для користувачів на додаткових умовах. Чим більше значення показника «квантовий об'єм» квантового комп'ютера, тим більшого розміру схему можна реалізувати на його кубітах.

Таблиця 3. Основні характеристики квантових симуляторів IBM

Назва квантового комп'ютера IBM	Кількість кубітів	«Квантовий об'єм»	Доступні квантові логічні вентиля	Вільний доступ
ibmq_16_melbourne	15	8	CX, ID, RZ, SX, X	Так

Продовження таблиці 3

Назва квантового комп'ютера ІВМ	Кількість кубітів	«Квантовий об'єм»	Доступні квантові логічні вентилі	Вільний доступ
ibmq_5_yorktown	5	8	CX, ID, RZ, SX, X	Так
ibmq_armonk	1	1	ID, RZ, SX, X	Так
ibmq_athens	5	32	CX, ID, RZ, SX, X	Так
ibmq_belem	5	16	CX, ID, RZ, SX, X	Так
ibmq_bogota	5	32	CX, ID, RZ, SX, X	Так
ibmq_brooklyn	65	32	CX, ID, RZ, SX, X	Ні
ibmq_cairo	27	64	CX, ID, RZ, SX, X	Ні
ibmq_casablanca	7	32	CX, ID, RZ, SX, X	Ні
ibmq_dublin	27	64	CX, ID, RZ, SX, X	Ні
ibmq_guadalupe	16	32	CX, ID, RZ, SX, X	Ні
ibmq_hanoi	27	64	CX, ID, RZ, SX, X	Ні
ibmq_jakarta	7	16	CX, ID, RZ, SX, X	Ні
ibmq_kolkata	27	128	CX, ID, RZ, SX, X	Ні
ibmq_lagos	7	32	CX, ID, RZ, SX, X	Ні
ibmq_lima	5	8	CX, ID, RZ, SX, X	Так
ibmq_manhattan	65	32	CX, ID, RZ, SX, X	Ні
ibmq_manila	5	32	CX, ID, RZ, SX, X	Так
ibmq_montreal	27	128	CX, ID, RZ, SX, X	Ні
ibmq_mumbai	27	128	CX, ID, RZ, SX, X	Ні
ibmq_nairobi	7	32	CX, ID, RZ, SX, X	Ні
ibmq_paris	27	32	CX, ID, RZ, SX, X	Ні
ibmq_peekskill	27	–	CX, ID, RZ, SX, X	Ні
ibmq_quito	5	16	CX, ID, RZ, SX, X	Так
ibmq_rome	5	32	CX, ID, RZ, SX, X	Ні
ibmq_santiago	5	32	CX, ID, RZ, SX, X	Так
ibmq_sydney	27	32	CX, ID, RZ, SX, X	Ні
ibmq_toronto	27	32	CX, ID, RZ, SX, X	Ні

ІВМ Quantum надає можливість будувати та реалізовувати квантові алгоритми у двох режимах – графічному та текстовому – за допомогою сервісів *IBM Quantum Composer* та *IBM Quantum Lab* відповідно.

IBM Quantum Composer [5] є найпростішим інструментом для створення, графічної візуалізації та запуску квантових схем на квантових симуляторах і на реальних квантових комп'ютерах ІВМ (Рис. 5).

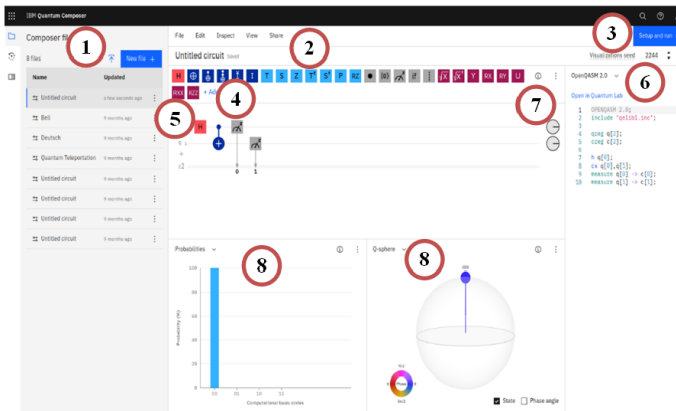



Рис. 5. IBM Quantum Composer

На рис. 5 показано:

1 — бічна панель, що надає доступ до власних файлів, завдань або документації. Відкрити або закрити бічну панель можна натисканням піктограми на вкладці;

2 — рядок меню, що використовується для створення нової схеми, керування та збереження схем, налаштування робочої області, отримання допомоги тощо;

3 — область входу в обліковий запис і налаштування параметрів для запуску квантової схеми;

4 — квантові вентиля та панель операцій. Різні типи вентилів (Таблиця 4) згруповані за кольором: класичні вентиля мають темно-синій колір, фазові вентиля — світло-блакитний, а неунітарні операції — сірий. Кнопка  дозволяє відкрити довідник із квантових операцій і вентилів, отримати допомогу щодо використання гарячих клавіш, згорнути панель квантових операцій до одного ряду, зберегти створену квантову схему як файл у різних форматах (pdf, svg, png);

5 — графічний редактор квантових схем. Додавання операцій, які будуть виконуватися над кубітами, здійснюється простим перетягуванням позначення вентиля в область редактора графічних квантових схем;







6 — редактор коду дає змогу переглядати та копіювати автоматично згенерований, на основі побудованої схеми, код у режимах OpenQASM або Qiskit для використання в інших програмах;

7 — фазові диски відображають стан кубіту як вектор у комплексній

площині, що задається радіальною лінією, яка обертається проти годинникової стрілки;

8 — візуалізації стану кубітів, що моделюють створювану схему у процесі побудови.

Таблиця 4. Деякі позначення, що застосовуються у IBM Quantum Composer

Елемент схеми	Позначення в IBM Quantum Composer	Приклад використання Quantum в Quantum Lab	Матричне подання
Вентиль H		<code>circuit.h(qreg)</code>	$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Вентиль X		<code>circuit.x(qreg)</code>	$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Вентиль CNOT (CX)		<code>circuit.cx(qreg)</code>	$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$
Вентиль Z		<code>circuit.z(qreg)</code>	$Z = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Вимірювач		<code>circuit.measure(qreg, creg)</code>	
Розмежовувач		<code>circuit.barrier()</code>	

Елементи, подані у таблиці 4:

- вентиль H, або вентиль Адамара, необхідний для переведення кубіту в стан суперпозиції;
- вентиль Паулі X еквівалентний бітовому запереченню;
- вентиль CNOT, також відомий як вентиль контрольованого заперечення (CX), діє на пару кубітів, один з яких контрольний, а інший — цільовий. Він виконує заперечення на цільовому кубіті кожного разу, коли контрольний кубіт дорівнює 1. Якщо контрольний кубіт знаходиться в суперпозиції, цей вентиль створює заплутування;

- вентиль Z змінює знак (напряв) кубіту;
- вимірювання кубітів — незворотня операція, що змінює стан кубіту. Результатом вимірювання є класичний біт;
- розмежовувач (бар'єр) корисний для візуалізації квантових схем.

У IBM Quantum Composer передбачена можливість візуалізувати схему квантових операції та результат у вигляді ймовірностей (probabilities), вектора стану на виході (statevector), на Q -сфері (не сфері Блоха), а також переглянути опис створеної схеми на квантовому асемблері QASM 2.0 або Qiskit з можливістю відкрити код у IBM Quantum Lab. Перегляд створеної схеми в режимі інспектування (Inspect) дає можливість спостерігати за покроковим виконання квантового алгоритму та зміною станів задіяних кубітів.

Для використання додаткових можливостей від IBM Quantum Composer — вибору симулятора або реального квантового комп'ютера — необхідно авторизуватися за даними існуючого власного облікового запису (Google, GitHub, Twitter, LinkedIn, Fraunhofer або електронною поштою) або зареєстрованого облікового запису IBM (із доступом до пробних версій, демонстраційних програм, стартових наборів, послуг та API). На додаткових умовах організаціям-партнерам проєкту IBM Quantum Network можна отримати доступ до найновіших систем квантових обчислень й інструментів розробки після подання електронної заявки та її розгляду [9].

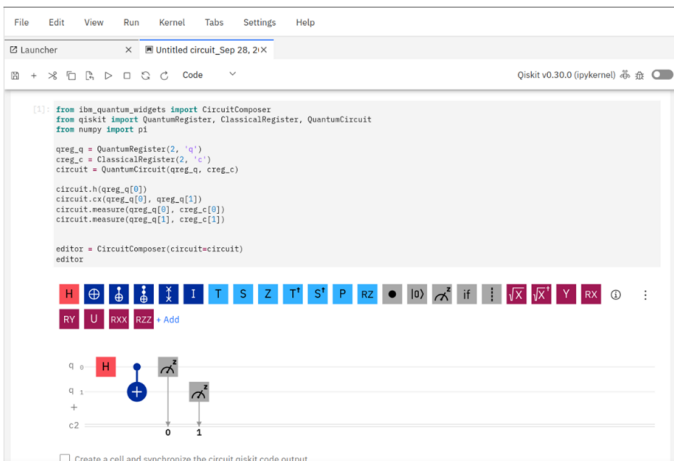


Рис. 6. Фрагмент сторінки сервісу IBM Quantum Lab

IBM Quantum Lab (<https://lab.quantum-computing.ibm.com>) — сервіс для реалізації квантових алгоритмів мовою програмування, тобто в текстовому режимі, виключно для авторизованих користувачів. За допомогою *IBM Quantum Lab* можна створити нову програму для квантового обладнання або відкрити автоматично згенерований програмний код квантової схеми, що раніше була побудована в *IBM Quantum Composer*. Інтерфейс *IBM Quantum Lab* буде знайомим тим користувачам, хто має досвід роботи з *Jupyter Notebook* (Рис. 6).

Запуск програмного коду відбувається звичними для користувачів *Jupyter Notebook* способами: кнопкою *Run* або комбінацією клавіш *Shift+Enter*. Під час побудови програми для квантового обладнання треба вказати необхідну кількість кубітів і класичних бітів (за замовчуванням кожний кубіт встановлюється в нульовий початковий стан). Потім слід додати вентиля для маніпулювання кубітами та вивести результат або додати спосіб візуалізації побудованої квантової схеми.

Можливості побудови, реалізації та запуску квантових алгоритмів на платформі *IBM Quantum* у графічному та текстовому режимах продемонструємо на прикладі *алгоритму квантової телепортації*.

Квантова телепортація — це передавання квантових станів від одного кубіту до іншого. Квантова телепортація не є транспортуванням або будь-яким фізичним переміщенням кубіту з одного місця в інше. У квантовій механіці діє теорема про заперечення клонування (копіювання) [19, с. 89]. У разі копіювання під час роботи на квантовому обладнанні відбувається неявне вимірювання, що руйнує поточний квантовий стан. Для вирішення цієї проблеми використовують явище *квантової заплутаності* — взаємозалежності квантових мікрочастинок (у цьому випадку — кубітів). Вимірювання стану одного заплутаного кубіту призводить до миттєвого переходу у відповідний стан іншого заплутаного кубіту.

Наведемо словесний опис алгоритму квантової телепортації, графічний опис якого здійснений за допомогою сервісу *IBM Quantum Composer*, що представлений на рисунку 7:

- 1) за допомогою операції *NOT* переведемо нульовий кубіт у стан 1, а перший і другий кубіти залишимо в первинному нульовому стані. Слід зауважити, що ця операція є обов'язковою для розглядуваного прикладу, виключно для того, щоб уникнути передавання нульового значення кубіту. Насправді, нульовий кубіт буде містити те значення, яке необхідно телепортувати;

- 2) переведемо перший кубіт у суперпозицію вентилям Н;
- 3) заплутаємо перший і другий кубіти вентилям CNOT (де перший — контрольний, а другий — цільовий. Якщо контрольний (перший) кубіт у стані 1, то цільовий (другий) інвертується вентилям CNOT);
- 4) аналогічно заплутаємо нульовий і перший кубіти;
- 5) переведемо нульовий кубіт у суперпозицію (за допомогою вентиля Н);
- 6) виміряємо стани нульового та першого кубітів (операція Measurement). Результати вимірювання збережемо у два класичні біти, що передаються звичайним (класичним, неквантовим) способом комунікації (каналом, протоколом);
- 7) на стороні, куди передається стан нульового кубіту, є другий кубіт, до якого застосовуємо вентиля СХ та СZ (у послідовності виконання СХ або СZ не має значення, що буде першим), у результаті отримаємо у другому кубіті значення нульового кубіту;
- 8) виміряємо значення другого кубіту.

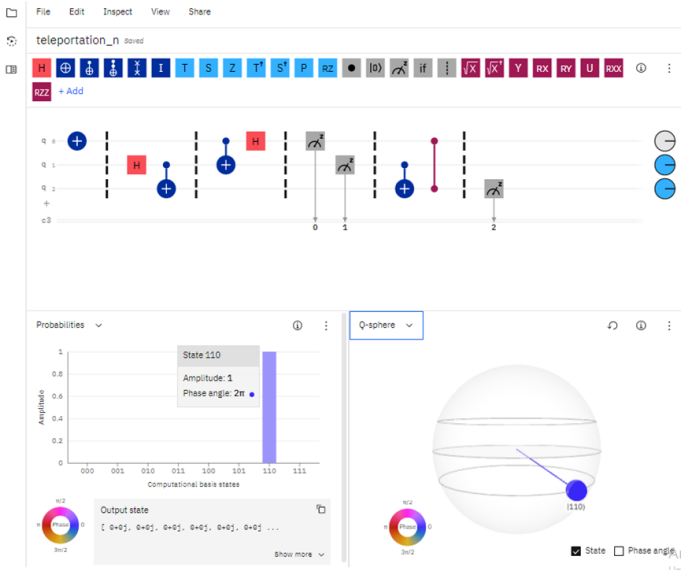


Рис. 7. Схема реалізації алгоритму квантової телепортації у IBM Quantum Composer

Проаналізуємо вигляд фазових дисків після побудови та запуску квантової схеми (Рис. 7):

- для нульового кубіту фазовий диск незафарбований, значить знаходиться у стані $|0\rangle$;
- для першого і другого кубітів фазовий диск зафарбований повністю, значить знаходиться у стані $|1\rangle$.

Розглянемо візуалізацію результату в режимі вектора стану, читаємо справа наліво — 011. Три позиції отриманого двійкового числа показують, що останній кубіт має значення 1.

Наведемо програмний код реалізації алгоритму квантової телепортації мовою Python.

```
# підключення модулів для роботи з квантовими алгоритмами
from qiskit import QuantumCircuit, transpile, Aer, IBMQ
from qiskit.tools.jupyter import *
from qiskit.visualization import *
from ibm_quantum_widgets import *
from numpy import pi
from qiskit import QuantumRegister, ClassicalRegister, |
QuantumCircuit

# ініціалізація квантового регістру для роботи з трьома
# кубітами
qreg_q = QuantumRegister(3, 'q')
# ініціалізація класичного регістру для роботи з трьома
# бітами
creg_c = ClassicalRegister(3, 'c')
# ініціалізація квантової схеми
circuit = QuantumCircuit(qreg_q, creg_c)

# перший крок алгоритму квантової телепортації
circuit.x(qreg_q[0])
circuit.barrier(qreg_q[0], qreg_q[1], qreg_q[2])

# другий крок алгоритму квантової телепортації
circuit.h(qreg_q[1])

# третій крок алгоритму квантової телепортації
circuit.cx(qreg_q[1], qreg_q[2])
circuit.barrier(qreg_q[1], qreg_q[0], qreg_q[2])
```



```
# четвертий крок алгоритму квантової телепортації
circuit.cx(qreg_q[0], qreg_q[1])

# п'ятий крок алгоритму квантової телепортації
circuit.h(qreg_q[0])
circuit.barrier(qreg_q[0], qreg_q[1], qreg_q[2])

# шостий крок алгоритму квантової телепортації
circuit.measure(qreg_q[0], creg_c[0])
circuit.measure(qreg_q[1], creg_c[1])
circuit.barrier(qreg_q[1], qreg_q[0], qreg_q[2])

# сьомий крок алгоритму квантової телепортації
circuit.cx(qreg_q[1], qreg_q[2])
circuit.cz(qreg_q[0], qreg_q[2])
circuit.barrier(qreg_q[1], qreg_q[0], qreg_q[2])

# восьмий крок алгоритму квантової телепортації
circuit.measure(qreg_q[2], creg_c[2])

# будуємо схему (спосіб 1)
editor = CircuitComposer(circuit=circuit)
editor

# будуємо схему (спосіб 2)
%matplotlib
circuit.draw(output='mpl')

# ініціалізація виконання схеми на симуляторі
simulator = Aer.get_backend('qasm_simulator')
# виконання схеми на ініціалізованому симуляторі
result = execute(circuit, backend=simulator, |
shots=1024).result()
# відображення результатів у вигляді гістограми
from qiskit.visualization import plot_histogram
plot_histogram(result.get_counts(circuit))
```

Наступні рядки програмного коду забезпечують виконання схеми на реальному квантовому комп'ютері після підключення облікового

запису входу до IBM Quantum Lab:

```
IBMQ.load\_account();  
provider = IBM.get\_provider(hub='ibm-q')  
qcomp = provider.get\_backend('ibmq\_16\_melbourn')  
import numpy as np
```

3 Висновки

Визначено критерії добору хмаро орієнтованих платформ для роботи з квантовими алгоритмами та навчання основ квантової інформатики учнів ліцеїв. На основі визначених критеріїв проведено порівняльний аналіз найпоширеніших хмаро орієнтованих платформ для квантових обчислень: Azure Quantum, Quantum Inspire, Amazon Braket, IBM Quantum. Здійснивши порівняльний аналіз згаданих платформ, було прийнято рішення рекомендувати організацію навчання основ квантової інформатики учнів ліцеїв на платформі IBM Quantum. Наразі відсутність україномовної локалізації інтерфейсу IBM Quantum не є суттєвою проблемою для учнів у 10–11 класах. За такої організації навчання природним є формування ключової багатомовної компетентності. Через стрімкий розвиток платформи IBM Quantum можна сподіватися на появу адаптивного дизайну та мобільного застосунку.

References

1. Bazurin, V.: Programming environments as a means of teaching students the basics of programming. *Information technologies and teaching tools* **59** (3), 13–27. http://nbuv.gov.ua/UJRN/ITZN_2017_59_3_4 (2017). Accessed 20 Jan 2021
2. Decree of the President of Ukraine No.126/2018 of 14.05.2018 «On the decision of the National Security and Defense Council of Ukraine of May 2, 2018 “On the application and cancellation of personal Special Economic and other restrictive measures (sanctions)”». <https://zakon.rada.gov.ua/go/126/2018> (2018). Accessed 20 Jan 2021
3. Decree of the President of Ukraine No.133/2017 of 15.05.2017 «On the decision of the national security and Defense Council of Ukraine of April 28, 2017 “On the application of personal Special Economic and other restrictive measures (sanctions)”». <https://zakon.rada.gov.ua/go/133/2017> (2017). Accessed 20 Jan 2021

4. Five years ago today, we put the first quantum computer on the cloud. Here's how we did it. https://research.ibm.com/blog/quantum-five-years?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000040UV&utm_term=10014622&utm_id=HabraRU-VSeredina-IBMQuantim-in-Cloud-2021-06-02 (2021). Accessed 20 Jan 2021
5. IBM Quantum Composer. <https://quantum-computing.ibm.com/composer/files/new> (2021). Accessed 20 Jan 2021
6. IBM Quantum Services Simulators. <https://quantum-computing.ibm.com/services?services=simulators> (2021). Accessed 20 Jan 2021
7. IBM Quantum Services System. <https://quantum-computing.ibm.com/services?services=systems> (2021). Accessed 20 Jan 2021
8. IBM Quantum. <https://quantum-computing.ibm.com> (2021). Accessed 20 Jan 2021
9. IT Infrastructure | IBM. <https://www.ibm.com/it-infrastructure/us-en/resources/campaignmail/quantum-computing/contact> (2021). Accessed 20 Jan 2021
10. Law of Ukraine “On copyright and related rights” of 23.12.1993 No. 3792-XII. <https://zakon.rada.gov.ua/go/3792-12> (2021). Accessed 20 Jan 2021
11. Lehka, L., Shokaliuk, S., Bohunenko, E.: Propaedeutics of studying quantum computer science in a specialized (high) school. *Physical and Mathematical Education* **28** (2), 51–56. <https://doi.org/10.31110/2413-1571-2021-028-2-009> (2021). Accessed 20 Jan 2021
12. Ministry of Education and Science of Ukraine — Curricula for grades 10–11. <https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/navchalni-programi-dlya-10-11-klasiv> (2017). Accessed 20 Jan 2021
13. Overview Microsoft Quantum. <https://azure.microsoft.com/ru-ru/solutions/quantum-computing> (2021). Accessed 20 Jan 2021
14. Qiskit. <https://qiskit.org> (2021). Accessed 20 Jan 2021
15. Quantum Computing Service-Amazon Braket-Amazon Web Services. https://aws.amazon.com/braket/?nc1=h_ls (2021). Accessed 20 Jan 2021
16. Quantum Inspire. <https://www.quantum-inspire.com/features> (2021). Accessed 20 Jan 2021

17. Resolution of the Cabinet of Ministers of Ukraine No. 1555 of October 13, 2000 “On approval of regulations on the distribution of copies of audiovisual works, volumes, videograms, computer programs, databases” as amended. <https://zakon.rada.gov.ua/go/1555-2000-%D0%BF> (2021). Accessed 20 Jan 2021
18. Shevchuk, P.: The problem of choosing a programming language and environment learning tool, reporting scientific conference. Institute of information technologies and teaching tools of the Academy of Medical Sciences of Ukraine, pp. 30–31, Kiev (2010)
19. Silva V.: Practical Quantum Computing for developer. St. Petersburg, Piter (2020)
20. Vakaliuk T. A.: Features cloud oriented learning environment for training bachelors information. Scientific journal of the national research university named after M. P. Dragomanov. Series 2. Computer-oriented learning systems **19** (26), 154–157. <https://sj.npu.edu.ua/index.php/kosn/article/view/28> (2019). Accessed 20 Jan 2021
21. Vorozhbyt A. V.: Using web-based technologies in learning Informatics in the establishments of general secondary education. Dissertation, National Pedagogical Dragomanov University. https://old.npu.edu.ua/images/file/vidil_aspirant/dicer/%D0%94_26.053.03/Vorozhbyt.pdf (2019). Accessed 20 Jan 2021
22. Yatsenko, A. I., Yatsenko, A. S.: Criteria and indicators for selecting the playing environment for the development of information and communication competence of future primary school teachers. Actual issues of modern computer science **12**, 105–108. http://eprints.zu.edu.ua/31178/1/Zbirnuk_Zhytomyr_IKT_2019-2.pdf (2019). Accessed 20 Jan 2021