

СЕКЦІЯ: ОСОБЛИВОСТІ СВІТОВИХ ТА ВІТЧИЗНЯНИХ ОСВІТНІХ СТРАТЕГІЙ ПІДГОТОВКИ IT-ФАХІВЦІВ

PROCESSING ЯК СЕРЕДОВИЩЕ ДЛЯ НАВЧАННЯ КРЕАТИВНОГО ПРОГРАМУВАННЯ

Бугаєць Наталія Олександрівна

кандидат педагогічних наук,

старший викладач кафедри інформаційних технологій і аналізу даних,

Ніжинський державний університет імені Миколи Гоголя

anatashika@gmail.com

Одним із додаткових модулів навчальної програми з інформатики для учнів старшої школи загальноосвітніх шкіл, рівень стандарту, є «Креативне програмування». Зміст навчання за цим модулем складається з таких тем: цифрове мистецтво та творчість, графічні побудови та взаємодії, функції, об'єкти та класи, мультимедіа, інтерфейс програмного продукту.

Креативне програмування означається як процес, що базується на феномені відкриття, який містить в собі дослідження, повторення і рефлексію, використання коду як основного «посередника» для одержання артефактів мистецтва [5]. Креативність – це діяльність з використанням уяви для створення чогось якісно нового. Вона формується, коли виникає можливість подумати над новою ідеєю, знайти спосіб, підхід до розв'язування проблеми [6].

Processing – це програмний засіб для креативного програмування, що використовується для роботи в галузі генеративного дизайну.

Генеративний дизайн – розробка дизайну методами програмування. Поняття «генеративний дизайн» – це напрям проектної діяльності на основі методу синтезу, який реалізується в різних галузях дизайну (мистецтво, архітектура, комунікації, промисловий дизайн) [2]. Завдяки можливостям опрацьовувати великі масиви даних за допомогою комп'ютера, дизайнер-програміст може визначити форми і об'єкти за рахунок створення і варіювання параметрів і алгоритмів.

Під час зміни параметрів можуть створюватися нові елементи і форми дизайну, які з'являються в процесі програмування. Оцінюючи результати і зміни в процесі, виконуючи доопрацювання, перетворення програмного коду, дизайнер-програміст вдосконалює процес і остаточний результат. Під час цього процесу в дизайнера-програміста може з'явитися значна кількість результатів, які можуть миттєво змінюватися зі зміною параметрів, а фактори випадковості і варіативності зазвичай приводять до цікавих креативних результатів.

Генеративне моделювання досить часто базується на ґрунтовних теоретичних основах математики та інформатики і потребує практичного досвіду. В першу чергу мова йде про так звані еволюційні алгоритми, в основі яких лежать математичні моделі механізмів природної еволюції. Серед них найбільш

розповсюджені методи: фрактали, L-системи, множина Мандельброта, шум Перліна, філотаксис, теорія хаосу, рандомізація, діаграма Вороного та ін. [2].

Генеративний дизайн використовується у всіх сферах, що пов'язані з тими чи іншими розробками і проектуванням. Наприклад, у промисловому дизайні використовуються численні найновіші засоби автоматичного проектування, які використовуються для оптимізації виготовлення виробів, зниження їх ваги і економії витратних матеріалів [2].

Комп'ютерні технології стали причиною появи нової галузі у творчості – цифрове мистецтво. Сучасні дизайнери одержали можливість створювати власні інструменти за допомогою мов програмування [1].

Науковець П. Орлов зазначає, що для креативного програмування характерні певні властивості, перша з яких пов'язана з особливостями творчого процесу: часто дизайнер починає творити, маючи лише задум, а не детально розроблене завдання, і результат може виявитися неочікуваним навіть для самого творця [5]. Отже, перша властивість креативного програмування полягає у відсутності формалізованого творчого завдання. Друга властивість пов'язана з можливістю створення індивідуальних художніх засобів за допомогою мов програмування.

За допомогою засобів Processing легко створювати зображення, анімацію, інтерактивну графіку. Вивчення Processing – це один із способів вивчення програмування на основі створення графічних проектів і візуалізації. Також вивчення мови Processing відкриває світ комп'ютерної графіки для тих, хто вже має деякі навички програмування. Це мова програмування, яку зручно використовувати для навчання програмування студентів, які цікавляться дизайном, а студентам технічних та комп'ютерних спеціальностей дає більш легкий інструмент для роботи з графікою.

За допомогою Processing є можливість вчитися програмуванню через написання інтерактивних програм. Такий шлях освоєння програмування є цікавим і мотивуючим для учнів старшої школи, оскільки програмування в Processing дає миттєвий результат, всі абстракції стають видимими, алгоритмічні конструкції розглядаються як практичні елементи втілення креативних ідей та творчих експериментів. З Processing програмування стає цікавою та креативною діяльністю, що заохочує до навчання та творчості.

Processing – це діалект мови програмування Java, доповнений спеціальними командами для роботи з графікою і зовнішніми пристроями. Графічні об'єкти Processing споріднені із системою PostScript і Open GL [4]. Також Processing має особливості, характерні для багатьох мов програмування і тому може слугувати гарним вступом в програмування на інших мовах з використанням інших засобів розробки (рис. 1).

Як і будь-яке програмне забезпечення, Processing складається з великої кількості компонентів, що працюють разом. Існує більше 100 бібліотек розширення, за допомогою яких можна застосовувати Processing для опрацювання звуку, досліджень машинного «зору», технічних розрахунків б].

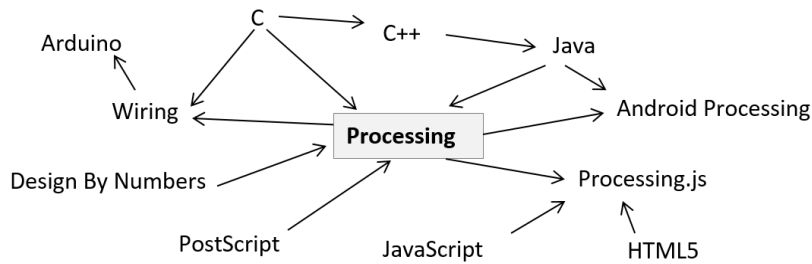


Рис. 1 Спорідненість Processing з іншими мовами і засобами розробки

Проект Processing був ініційований Бенджаміном Фраєм і Кейсі Різом в Media Lab Массачусетського технологічного інституту і до цього часу активно розвивається некомерційною ініціативною групою (Processing Foundation) [4].

Програма Processing розроблена за принципом FLOSS (Free Libre Open Source Software). Її можна безкоштовно завантажити із сайту <https://processing.org>. Також кожен може знайти та поділитися своїми цікавими проектами і знаннями на сайтах, що присвячені особливостям роботи в Processing (<https://funprogramming.org/>, www.openprocessing.org/).

Для мови креативного програмування Processing характерні такі особливості: простий і зрозумілий синтаксис мови; можливість працювати в режимі реального часу (live coding), коли можна бачити результат програми одразу після введення рядка з програмним кодом у редакторі; простота установки програми на платформах різних операційних систем; повна «комплектація» мови, включаючи управління сторонніми бібліотеками і сумісність версії для розв'язування графічних, художніх завдань [6].

Processing успішно використовується в освітньому середовищі, оскільки процес втілення художніх творчих задумів іде швидко, просто і разом з тим учні та студенти вчаться програмувати, використовувати базові алгоритмічні конструкції, системні змінні, функції, рекурсивні алгоритми, застосовувати об'єктно-орієнтований підхід у програмуванні, опрацьовувати мультимедійні дані у візуальному контексті. Слід зазначити, що для програмування в Processing необхідні базові знання математики, зокрема геометрії, що сприяє розвитку математичних компетентностей учнів і студентів, формує розуміння міжпредметних зв'язків математичних, інформатичних дисциплін і їх зв'язок із сучасним цифровим мистецтвом та дизайном.

Список використаних джерел:

1. Генеративний дизайн: на перетині мистецтва й програмування. URL: <http://telegraf.design/generativnij-dizajn-na-peretini-mistetstva-j-programuvannya/> (дата звернення 03.04.2019).
2. Генеративний дизайн: на порозі нової епохи проектування. Набр: веб-сайт. URL: <https://habr.com/ru/company/nanosoft/blog/345500/> (дата звернення 03.04.2019).
3. Гласс Р. Креативное программирование 2.0.: Пер. с англ. СПб.: Символ-Плюс, 2009. 352 с.
4. Кейси Риз, Бен Фрай. Учимся программировать вместе с Processing. Published by O'Reilly Media, Inc. Gravenstein Highway North, Sebastopol, 2010. 209 с.
5. Орлов П. А. Программирование для дизайнеров: учеб. пособие; под ред. проф. В.М. Иванова. М.: АВАТАР, 2015. 247 с.

6. Ян Вантомм. Processing 2: креативное программирование. Издательство: Published by Packt Publishing Ltd., 2012. 292 с.

7. Processing. Environment (IDE). URL: <https://processing.org/reference/environment/> (дата звернення 03.04.2019).

ОГЛЯД СЕРЕДОВИЩА JUPYTER ДЛЯ ВИВЧЕННЯ МОВИ ПРОГРАМУВАННЯ PYTHON

Войтенко Альона Ярославівна

магістрант спеціальності «Середня освіта. Інформатика»

Тернопільський національний педагогічний університет імені Володимира Гнатюка

Габрусєв Валерій Юрійович

кандидат педагогічних наук, доцент кафедри інформатики та методики її навчання
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
gabrusev@fizmat.tnpu.edu.ua

На сьогоднішній день у зв'язку із швидким розвитком комп'ютерної техніки, відбувається поступова зміна вимог до навчання основ програмування. Найперше питання, яке виникає перед початком навчання програмуванню, це вибір мови програмування та відповідних інструментальних засобів для цієї мови. Це питання немає однозначної відповіді, тому що залежить від багато факторів, зокрема від мети навчання, галузі застосування отриманих знань, середовищ розробки, популярності у професійних розробників тощо. Все залежить від людини та сфери діяльності, в якій вона планує застосовувати свої знання та навички [2].

На даний час великий інтерес представляє мова програмування Python. Вона впевнено набирає популярності за рахунок прозорого синтаксису та універсальності застосувань. Python – багатоцільова мова програмування, яка дозволяє писати код, який добре читається. Відносний лаконізм мови надає можливості створювати програми, які будуть набагато коротшими аналогу написаного на іншій мові програмування [3], а значна кількість різнопланових бібліотек практично знімає обмеження на галузі розробки програмного забезпечення. За допомогою мови програмування Python можна розробляти інструменти для супроводу інформаційної структури підприємства, універсальні додатки із графічним інтерфейсом, створювати ігри, веб-додатки, проводити дослідження у галузі опрацювання великих масивів даних та штучного інтелекту.

Під час розробки нескладного програмного забезпечення достатньо простого текстового редактора, але ситуація змінюється під час складних програмних проєктів. У цьому випадку виникає потреба у використанні інтегрованих середовищ розробки (IDE) та інших додаткових інструментальних засобів інтегрованих у IDE. Ефективність роботи залежить від правильного вибору такого середовища [1].

Серед усього загалу інструментальних засобів розробки мовою для мови програмування Python, доцільно розглянути Jupyter та JupyterHub.