

достаточно много. Поэтому возникает возможность проведения лекционных и практических занятий непосредственно на территории ИТ-компании, что позволяет студенту посмотреть на мир программистов изнутри. С другой стороны, преподаватели от производства активно отбирают студентов для прохождения практики в компании с дальнейшей возможностью трудоустройства.

Модель образования ВУЗ+ИТ-компания предполагает самостоятельность обучения и мотивированность в получении образования. В модели должны быть реализованы основные тренды современного образования: образование в течение всей жизни, использование дистанционного образования, открытость образования. Предполагается широкое использование технологий облачного образования.

Вуз закладывает у студентов базовые и теоретические знания и готовность к обучению в течение всей трудовой жизни. При этом специалист может 4-5 раз поменять профессию. Раннее появление на производстве позволяет получить быстрое вхождение обучаемого студента в производственные проблемы и задачи и быстрый карьерный рост.

## **ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ В ЯЗЫКАХ JAVA И SCALA**

**Блинов И. Н., Романчик В. С.**

*Белорусский государственный университет, Минск, Беларусь*

*e-mail: romanchikvs@gmail.com, blinov@gmail.com*

Технология объектно-ориентированного программирования (ООП) стала популярной, начиная с 90-х годов прошлого века и используется до нашего времени вместе с языками C++, Object Pascal, Java. Проблемы с ООП начались с широким использованием распределенных, параллельных и многопоточных парадигм программирования. ООП относится к стилю с изменяемым состоянием как в императивных языках, которые написаны в ООП стиле. Вызываемые методы объекта при этом предполагают возможность изменять через ссылку текущие состояния объектов.

**Функциональное программирование** — это стиль написания программ через использование набора функций. Основной принцип ФП — оборачивать практически все в функции, писать множество маленьких многократно используемых функций, а затем просто вызывать их одну за другой, чтобы получить результат вроде `f1(f2(f3))`.

В функциональных языках программирования есть и ряд ограничений, в частности — инвариантность данных. Инвариантные функции принимают аргументы и возвращают результат, не изменяя окружения (передача аргументов по значению). Поэтому ими значительно проще пользоваться при параллельном программировании.

Основная идея функционального программирования заключается в возможности создания функций как объектов, в частности, создание функции во время исполнения. Соответственно, программа может оперировать не только данными, но и функциями. Переменных не существует, даны только неизменяемые значения.

Особенности ФП:

1. Функции первого класса (First-Class Functions)
2. Функции высшего порядка (High-Order Functions)
3. Чистые функции (Pure Functions)
4. Замыкания (Closures)
5. Неизменяемое состояние (Immutable State)

До выхода Java 8 можно было создавать функции во время выполнения только одним способом: генерировать и загружать байт-код. Java8 и Scala уже имеют лямбда выражения.



В тоже время имеется изменяемое состояние самого объекта, но обеспечивается поддержка функционального программирования. Основная идея функционального программирования (ФП) получила развитие: используй изменяемое состояние, только тогда, когда это действительно необходимо.

Лямбда-выражения или подругому «замыкания» призваны упростить функциональное программирование. Для лямбда-выражений характерны две следующие черты:

1. Более лаконичный синтаксис при создании функций
2. Возможность создавать функцию, как объект, во время исполнения; затем эта функция в виде объекта может быть передана другому коду, либо другой код может ею оперировать.

Замыкания касаются второй возможности. При замыкании используются переменные, расположенные вне области действия функции. В традиционном процедурном программировании это не представляет проблемы — переменная просто используется — но проблема возникает, как только функция создается во время исполнения.

Лямбда-выражения описывают, какое действие должно быть выполнено, а не способ его выполнения. Имя функции остается неизвестным и реально простой пользователь может даже и не знать от о типе параметра метода, принимающего подобную функцию, и тем более не знать об имени самого метода ничего. Пользователь видит только необходимый для решения его задачи функционал, то есть код самого метода. В традиционном программировании использовалась бы внешняя итерация\цикл, в которой явно указывается всю последовательность операций, а также способ их выполнения, что увеличивало количество кода многократно. Конечно функциональный код короче, но и на первый взгляд он выглядит практически нечитабельным, что противоречит принципам предельной ясности языка Java. Это поднимает планку для вхождения в изучение языка. Но опытному программисту код ФП ясен и легко читаем и делает сам процесс разработки более быстрым и гибким к изменениям.

Лямбда-выражения в Java замыкаются только вокруг значений, но не вокруг переменных. Java требует, чтобы эти значения были неизменны, как бы объявлены как `final` независимо от того, действительно объявлены они таким образом или нет. То есть, переменные должны быть финальными по факту. Поэтому в Java есть «замыкания с ограничениями», а не «полноценные» замыкания, которые, тем не менее, довольно полезны.

При создании объектов, расположенных не в «куче», их можно изменять, поскольку компилятор следит лишь за тем, чтобы не изменялась сама ссылка. Например:

```
import java.util.function.Consumer;
class Info {
    String task;
}
class InfoClosure {
    Consumer<String> action() {
        Info info = new Info ();
        return o -> info.task += o;
    }
}
```

Метод `action()` в качестве возвращаемого значения использует функциональный интерфейс `Consumer<T>`. Возвращается таким образом объект, оборачивающий функцию. Роль этого объекта предоставить доступ к этой функциональности. Причем эта функциональность может быть в любой момент изменена программистом. Компиляция проходит успешно, чтобы в этом убедиться можно просто поставить ключевое слово `final` в объявлении объекта `task`. Если применить такой ход с любой конкурентным доступом, то возникнет проблема с изменяемым разделяемым состоянием.

Лямбда-выражения позволяют достичь желаемой цели: теперь можно создавать функции динамически. Причем функциональное API спроектировано так, что разработчику нет необходимости создавать свои собственные функциональные интерфейсы. Возможности пакета `java.util.function` практически полностью покрывают все возможно возникающие задачи. К тому же некоторые известные интерфейсы, а именно `Comparator`, `Runnable` и некоторые другие приобрели функциональный статус, что позволяет очень удобно их использовать и комбинировать.

Эти структуры названы «лямбдами», а не просто «замыканиями». «Замыкание» — неудачный и перегруженный термин.

Конфликта «ООП против ФП» в Java не возникает. ООП хорошо подходит и для абстрагирования данных и поведений как в ФП. Обе парадигмы полезны, и тем более полезны, если их смешивать в Java 8.

Проведенный обзор основан на подготовленных новых курсах и учебных пособиях по функциональному программированию на ММФ БГУ.

## SWOT-АНАЛИЗ ПРИМЕНЕНИЯ ДИСТАНЦИОННОГО ОБУЧЕНИЯ В РЕСПУБЛИКЕ БЕЛАРУСЬ

Богатко А. В., Богатко М. П.

Главный информационно-аналитический центр Министерства образования Республики  
Беларусь, Минск, Беларусь, e-mail: [sobol@unibel.by](mailto:sobol@unibel.by)

В Республике Беларусь все большее распространение получает дистанционное обучение (далее — ДО). Такое обучение направлено на обновление и углубление знаний, развитие профессиональных умений и навыков [1]. Однако неоднозначность применения ДО способствует возникновению необходимости выполнения анализа сильных (Strengths) и слабых (Weaknesses) сторон внутренней среды с точки зрения возможностей (Opportunities) и угроз (Threats) внешней среды (SWOT-анализ).

| Сильные стороны                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Слабые стороны                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> <li>1. Расширение масштабов подготовки кадров</li> <li>2. Доступность получения образования для населения</li> <li>3. Использование разнообразных методов донесения учебной информации</li> <li>4. Возможность рационального распределения времени в процессе обучения</li> <li>5. Наличие индивидуального подхода с учетом потребностей и психологических особенностей обучающихся</li> <li>6. Наличие обратной связи с преподавателем (в т.ч. индивидуальное общение)</li> <li>7. Гибкость в отношении того, как и когда получать образование</li> <li>8. Сокращение издержек получения образования</li> </ol> | <ol style="list-style-type: none"> <li>1. Отсутствие очного общения между обучающимися и преподавателем</li> <li>2. Высокая трудоемкость разработки курсов дистанционного обучения</li> <li>3. Необходимость наличия ряда индивидуально-психологических условий (в т.ч. самодисциплина)</li> <li>4. Высокая стоимость построения системы дистанционного обучения</li> <li>5. Отсутствие эффективных методов ориентирования в информационном пространстве Интернет</li> <li>6. Дистанционные программы не подходят для овладения профессиями, где необходимо много практики</li> </ol> |
| Возможности                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Угрозы                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <ol style="list-style-type: none"> <li>1. Повышение уровня образования населения за счет доступности получения образования</li> <li>2. Повышение экономических показателей страны за счет получения образования без отрыва от производства</li> <li>3. Возможность получения образования на базе ведущих учреждений образования</li> <li>4. Возможность реализации дополнительного образования взрослых, дополнительного образования детей и молодежи, специальное образование [2].</li> </ol>                                                                                                                                                      | <ol style="list-style-type: none"> <li>1. Высокая конкуренция с зарубежными учреждениями образования, предоставляющими дистанционное обучение в свободном доступе в Интернете</li> <li>2. Появление дистанционного обучения «низкого качества» в регионах</li> <li>3. Недостаточное количество аккредитованных и сертифицированных профессиональных курсов</li> </ol>                                                                                                                                                                                                                 |

Рис. 1. SWOT-анализ дистанционного обучения в Республике Беларусь