

УДК 372.8

**Возможности и использование комбинированного
типа данных в Pascal**

**Бегеза Е. В., студент
Клебча Е. Ю., студент**

*Белорусский национальный технический университет,
г. Минск, Республика Беларусь*

Научный руководитель: преподаватель Михасик Е. И.

Аннотация.

В данной научной работе рассмотрен комбинированный тип данных *Pascal*, а именно запись, его принцип работы и возможности, которые он раскрывает. Также в статье отображены варианты написания и использования кода с помощью записи, их плюсы и минусы для работы программиста. Благодаря этой работе вы глубже познакомитесь и изучите комбинированный тип данных в *Pascal*, а также узнаете о его преимуществах.

Для использования в программе множества однотипных данных используются массивы. Однако, очень часто возникает необходимость использовать разнородные типы данных для описания одного и того же объекта. В этом случае в программе используется тип данных, именуемый запись.

Другими словами, массивы *Pascal* позволяют определять тип переменных, которые могут содержать несколько элементов данных одного типа, но запись – это еще один определяемый пользователем тип данных, доступный в *Pascal*, который позволяет комбинировать элементы данных разных типов.

Запись – одна из самых полезных и отличительных особенностей языка *Pascal*. Он предоставляет средства для объединения набора различных типов данных в одну именованную структуру.

В *Pascal* запись является единственной концепцией структуры типа данных, которая позволяет изменять ее структуру во время выполнения программы. Это сверхпрактичное свойство записи позволяет написать универсальный код, охватывающий множество разнообразных случаев.

Записи состоят из разных полей. Предположим, необходимо отследить книги в библиотеке. Их можно отследить по следующим атрибутам каждой книги:

1. Заголовок.
2. Автор.
3. Предмет.
4. Идентификатор книги.

В типе данных запись разрешены поля варианта. Это значит, что несколько полей могут совместно использовать одну и ту же область памяти, а сами поля могут быть разных типов (это соответствует объединению соединений в языке C++). Другими словами, таким образом можно получить доступ к одной и той же ячейке памяти в записи через вариантное поле или набор полей, но каждое значение все равно необходимо обрабатывать по-разному.

Эта концепция открывает новые горизонты: возможность создавать программы более интерактивно и аккуратно. Теперь можно выбрать вариант на основе данных времени выполнения (данные, которые считываются во время работы программы). Поскольку в любой момент времени (после первого присвоения значения селектору вариантов) только один вариант является «активным», в программе произойдет сбой, если будет попытка прочитать/записать значения «активного» варианта. Это желательное поведение, так как в этом вся суть наличия различных вариантов. Это гарантирует полную целостность написанных программ.

Pascal также позволяет иметь анонимные варианты, то есть селекторы, не носящие никакого имени.

В таком случае программист не сможет явно выбрать (или запросить) какой-либо вариант, поэтому в свою очередь, все варианты считаются «активными» одновременно. Поскольку нет именованного селектора, программа не может отследить, какой вариант должен работать, а какой «бракованный». Поэтому именно программист несет ответственность за определение того, какой вариант он может разумно считывать/писать в настоящее время.

Анонимные варианты часто используются для реализации «приведения типов». Если есть анонимная вариантная часть, то программист может объявить элементы, несущие разные типы данных, кото-

рые, в свою очередь, определяют метод интерпретации базовых данных. Затем можно использовать тот факт, что многие компиляторы помещают все варианты в один и тот же блок памяти.

Таким образом, комбинированный тип данных в *Pascal* позволяет решать задачи, которые стандартными методами решить довольно проблематично, а иногда и невозможно.

Так, например, при работе с массивами основное ограничение заключается в том, что каждый элемент должен иметь один и тот же тип. Но при решении многих задач возникает необходимость хранить и обрабатывать совокупности данных различного типа.

Для этого и существуют записи. Их использование позволяет развить не только навыки программирования, но также алгоритмическое и творческое мышления.

Список использованных источников

1. Семакин, И. Г. Основы программирования / И. Г. Семакин, А. П. Шестаков. – М. : Мастерство, 2002. – 432 с.
2. Андреева, Т. А. Программирование на языке *Pascal* / Т. А. Андреева. – М. : Бином, Лаборатория знаний, 2013. – 240 с.
3. Кетков, Ю. Л. Свободное программное обеспечение. *FREE PASCAL* для студентов и школьников / Ю. Л. Кетков. – СПб : БХВ-Петербург, 2011. – 384 с.
4. Павловская Т. А. Паскаль. Программирование на языке высокого уровня / Т. А. Павловская. – СПб : Питер, 2012. – 393 с.
5. Семакин, И. Г. Основы алгоритмизации и программирования: учебник для ССУЗов / И. Г. Семакин, А. П. Шестаков. – М. : Академия, 2008. – 400 с.