Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы трансляции

ОТЧЁТ

к лабораторной работе №1

на тему

**ОПРЕДЕЛЕНИЕ МОДЕЛИ ЯЗЫКА**

Выполнил: студент гр.253505 Таргонский Д.А.

Проверил: ассистент кафедры информатики Гриценко Н.Ю.

Минск 2024

**СОДЕРЖАНИЕ**

[1 Цель работы 3](#_Toc188880349)

[2 Подмножество языка программирования 4](#_Toc188880350)

[3 Инструментальная языковая среда 5](#_Toc188880351)

[Заключение 6](#_Toc188880352)

[Список использованных источников 7](#_Toc188880353)

[Приложение А (обязательное) Исходный код программного продукта 8](#_Toc188880354)

# 1 ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является изучение процесса трансляции с одного языка программирования на другой, а именно создание транслятора, который преобразует код с языка *PHP* в код на языке *Visual* *Basic*. Этот транслятор реализуется с использованием языка *Visual* *Basic* в интегрированной среде разработки *Visual* *Studio* 2022. Основной задачей является определение подмножества языка *PHP*, которое включает ключевые элементы, необходимые для написания базовых программ, а также их дальнейшая конвертация в эквивалентный код на языке *Visual* *Basic*.

Необходимость определения подмножества языка обусловлена тем, что язык *PHP* обладает богатым набором синтаксических конструкций, типов данных и инструментов, но для реализации транслятора важно выбрать лишь те элементы, которые представляют собой основу языка. В рамках данной работы в подмножество языка *PHP* включаются числовые и текстовые константы, основные типы переменных, операторы циклов, условные операторы, функции и базовые структуры данных, включая массивы и объекты. Это подмножество является минимальным, но достаточным для написания программ, демонстрирующих основные принципы работы языка *PHP*. Используя данные подмножества, были составлены 3 программы на *PHP* для непосредственной трансляции.

Кроме того, важной частью работы является выбор инструментальной языковой среды для реализации транслятора. Это включает определение языка программирования, на котором создаётся транслятор, операционной системы для разработки, а также аппаратной платформы, на которой выполняются все эксперименты. Все эти элементы должны быть тщательно подобраны, чтобы обеспечить стабильную работу транслятора и удобство разработки

# 2 ПОДМНОЖЕСТВО ЯЗЫКА ПРОГРАММИРОВАНИЯ

Язык программирования *PHP* является одним из наиболее популярных инструментов для веб-разработки. Он широко используется для создания динамических веб-сайтов и веб-приложений благодаря своей простоте, гибкости и богатой библиотеке функций. *PHP* поддерживает множество типов данных, структур и операторов, что делает его универсальным инструментом для работы с информацией.

В таблице 1.1 представлено подмножество языка программирования *PHP*, включающее основные операторы, функции и структуры данных, которые являются фундаментальными для понимания и использования *PHP*. Эти элементы составляют основу языка и являются ключевыми для выполнения различных задач, от простых вычислений до сложных веб-приложений.

Таблица 1.1 – Подмножество языка *PHP*

|  |  |  |
| --- | --- | --- |
| № | Категория | Пример |
| 1 | Типы переменных | $*x* = 10; (целое число)  $*y* = 3.14; (число с плавающей точкой)  $*str* = "*Hello*, *World*!"; (строковая константа)  $*a* = *true*; (логический) |
| 2 | Операторы циклов | *for* ($*i* = 0; $*i* < 10; $*i*++) {}  *do* { } *while* ($*condition*); |
| 3 | Структуры данных | $*obj* = (*object*) *[*'*key*' => '*value*'*]*; (объект)  $*arr* = *[*1, 2, 3*]*; (массив) |
| 4 | Функции | *function multiply($a, $b) { return $a \* $b; }* |
| 5 | Условные операторы | *if* ($*x* > 0) { } *else* { }  *switch* ($*var*) { *case* 1: ... } |

Как видно из таблицы 1.1, данное подмножество охватывает основные элементы языка *PHP*, необходимые для выполнения широкого спектра задач.

# 3 ИНСТРУМЕНТАЛЬНАЯ ЯЗЫКОВАЯ СРЕДА

Для реализации транслятора используется язык программирования *Visual* *Basic*, который входит в состав платформы .*NET* *Framework*. *Visual* *Basic* был выбран за его высокую производительность, строгую типизацию и богатый набор инструментов для работы с текстовыми данными, что делает его подходящим для задачи трансляции. Разработка ведётся в интегрированной среде *Visual* *Studio* 2022, которая предоставляет мощные инструменты для написания, отладки и тестирования кода. *Visual* *Studio* включает встроенный отладчик, средства анализа кода и поддержку работы с системой контроля версий, что значительно упрощает процесс разработки.

Операционной системой, используемой для выполнения лабораторной работы, является *Windows* 11. Эта операционная система обеспечивает стабильную работу *Visual* *Studio* 2022 и предоставляет широкие возможности для настройки окружения разработки. Стабильность и совместимость *Windows* 11 с различными инструментами разработки делают её оптимальным выбором для выполнения данной работы.

Рабочей аппаратной платформой является персональный компьютер (ноутбук) с процессором *Intel* *Core* *i*3, 8 ГБ оперативной памяти и *SSD*-накопителем. Такое оборудование обеспечивает приемлемую производительность при работе с ресурсозатратными задачами, такими как разработка и тестирование транслятора. Используемый монитор с высоким разрешением позволяет комфортно работать с большим количеством кода, а периферийные устройства, такие как клавиатура и мышь, способствуют продуктивной работе.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы было проведено определение подмножества языка программирования *PHP*, включающего все основные элементы, необходимые для разработки программ. Были рассмотрены типы переменных, операторы цикла, структуры данных, функции, условные операторы. Это подмножество охватывает все стандартные возможности *PHP*, что позволяет использовать его для решения широкого круга задач, начиная от простых вычислений и заканчивая сложной обработкой данных.

Особое внимание было уделено выбору инструментальной среды. Выбранная среда обеспечивает стабильную работу всех компонентов языка и предоставляет удобные инструменты для разработки и тестирования программ.

Разработанные в рамках работы программы демонстрируют применение определённых элементов подмножества языка *PHP*. Они включают работу с числовыми и текстовыми константами, использование различных типов переменных, применение операторов цикла и условных операторов, а также манипуляции со структурами данных и функциями.

В заключение можно отметить, что выполнение данной лабораторной работы позволило не только изучить основные элементы языка *PHP*, но и получить практический опыт их применения. Это подмножество языка, а также выбранная инструментальная среда, могут быть успешно использованы для решения задач в области веб-разработки, анализа данных и других направлений. Полученные знания и навыки станут основой для дальнейшего изучения языка *PHP* и его применения в профессиональной деятельности.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] PHP Documentation. [Электронный ресурс]. – Режим доступа:  https://www.php.net/docs.php.

[2] Visual Basic documentation. [Электронный ресурс]. – Режим доступа : https://learn.microsoft.com/en-us/dotnet/visual-basic.

[3] Руководство по PHP. [Электронный ресурс]. – Режим доступа : https://metanit.com/php/tutorial.

[4] Руководство по языку программирования VB.NET. [Электронный ресурс]. – Режим доступа : https://metanit.com/visualbasic/tutorial.

# ПРИЛОЖЕНИЕ А

# (обязательное)

# Исходный код программного продукта

// Программа 1

<?php

// Пример использования циклов и условных операторов

for ($i = 1; $i <= 10; $i++) {

echo $i . "\n";

}

$x = 1;

while ($x < 5) {

echo $x . "\n";

$x++;

}

do {

echo $x . "\n";

$x++;

} while ($x < 7);

$number = 10;

if ($number > 0) {

echo "Число положительное\n";

} else {

echo "Число неположительное\n";

}

$fruit = "apple";

switch ($fruit) {

case "apple":

echo "Это яблоко\n";

break;

case "banana":

echo "Это банан\n";

break;

default:

echo "Неизвестный фрукт\n";

}

?>

// Программа 2

<?php

// Пример использования структур данных и функций

$vector\_example = array(1, 2, 3);

print\_r($vector\_example);

$assoc\_array = array("a" => 1, "b" => "text");

print\_r($assoc\_array);

function myFunction($x) {

return $x \* 2;

}

$result = myFunction(5);

echo $result . "\n";

?>

// Программа 3

<?php

// Пример использования различных типов переменных

$x = 10;

$y = "Hello";

$z = true;

$arr = array(1, 2, 3);

$obj = new stdClass();

if ($x > 5) {

echo "x больше 5\n";

} else {

echo "x меньше или равно 5\n";

}

if ($z) {

echo "Логическая переменная z истинна\n";

}

foreach ($arr as $value) {

echo "Элемент массива: $value\n";

}

echo "Тип переменной y: " . gettype($y) . "\n";

?>