Учреждение образования

 «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

**Отчет по лабораторной работе №1**

**по курсу ЯПИС**

**Вариант 19**

Выполнил студент группы 921702: Ладейко А.В.

Проверил: Баснина А.П.

МИНСК 2022

**Вариант:** 19

**Язык:** язык для работы с множествами

**Свойства:**

1. Объявление переменных: не явное.
2. Преобразование типов: явное, например, a = (int)b.
3. Оператор присваивания: Одноцелевой, например, a = b.
4. Структуры, ограничивающие область видимости: подпрограммы.
5. Маркер блочного оператора: не явный, например, как в python.
6. Условные операторы: двухвариантный оператор if-then-else.
7. Перегрузка подпрограмм: отсутствует.
8. Передача параметров в подпрограмму: только по значению и возвращаемому значению.
9. Допустимое место объявления подпрограмм: в начале программы.

**Целевой код:** Исполняемый файл (.exe), формат промежуточного кода – язык C++, генерация целевого кода стандартным компилятором.

**Этап №1**

**Объявление переменных:**

firstElement = element();

secondElement = element();

newSet = {firstElement, secondElement};

**Преобразование типов:**

elementForConversion = element();

newSet = (set) elementForConversion;

**Оператор присваивания:**

/\* Пример присваивания для element \*/

firstElemet = element();

secondElement = element();

firstElement = secondElement;

/\* Пример присваивания для set \*/

firstSet = {randomElement, randomElement2};

secondSet = {randomElement3, randomElement4};

firstSet = secondSet;

**Структуры, ограничивающие область видимости:**

function someName(element someElement):

newSet = {someElement};

return newSet;

…

firstElemet = element();

secondElement = element();

mySet = {firstElement, secondElement};

mySet = newSet; // Ошибка, множество newSet (function someName) недоступно!

**Маркер блочного оператора:**

firstElement = element();

secondElement = element();

while(firstElement == secondElement):

// участок кода в операторе while

**Условные операторы:**

firstElement = element();

secondElement = element();

if(firstElement == secondElement):

// участок кода, который отработает в случае, если firstElement == secondElement == true

else:

// участок кода, который отработает в случае, если если firstElement == secondElement == false

**Перегрузка подпрограмм:**

function someName(element firstElement, element secondElement):

// участок кода для данной функции

// попытка перегрузки подпрограммы someName, которая приведет к ошибке т.к перегрузка недопустима

function someName(element firstElement, element secondElement):

// участок кода для данной функции

**Передача параметров в подпрограмму:**

/\*Передача передаваемого параметра в подпрограмму по значению и возвращаемому значению\*/

function someName(element firstElement, element secondElement):

// участок кода для данной функции

**Допустимое место объявления подпрограмм:**

Реализация продемонстрировано в пункте «Структуры, ограничивающие область видимости»

**Перегрузка операторов для встроенных типов:**

**Для типа set:**

* **Объединение «+»:**

myFirstSet = {a, b, c}

mySecondSet = {d}

myThirdSet = myFirstSet + mySecondSet; // {a, b, c, d}

* **Разность «-»:**

myFirstSet = {a, b, c}

mySecondSet = {c}

myThirdSet = myFirstSet - mySecondSet; // {a, b}

* **Декартово произведение «\*»:**

myFirstSet = {a, b, c}

mySecondSet = {d}

myThirdSet = myFirstSet \* mySecondSet; // {{a, d}, {b, c}, {c ,d}}

* **Cимметрическая разность «/»:**

myFirstSet = {a, b, c}

mySecondSet = {c, d}

myThirdSet = myFirstSet / mySecondSet; // {a, b, d}

* **Оператор «больше» «>»:**

myFirstSet = {w, e, f, v, s, o}

mySecondSet = {a, b, c}

myFirstSet > mySecondSet; // true

* **Оператор «меньше» «<»:**

myFirstSet = {a, b, c}

mySecondSet = {w, e, f, v, s, o}

myFirstSet < mySecondSet; // true

* **Оператор «сравнение на равенство» «==»:**

myFirstSet = {a, b, c}

mySecondSet = {a, b, c}

myFirstSet == mySecondSet; // true

* **Оператор «сравнение на неравенство» «!=»:**

myFirstSet = {a, b, c}

mySecondSet = {a, b, s, d}

myFirstSet != mySecondSet; // true

**Для типа element:**

* **Оператор «сравнения на равенство» «==»:**

firstElement = element();

secondElement = firstElement;

firstElement == secondElement // true

* **Оператор «сравнение на неравенство» «!=»:**

firstElement = element();

secondElement = element();

firstElement != secondElement // true