

## Лабораторная работа № 2

### КЛАССЫ И ОБЪЕКТЫ JAVA

**Цель.** Получить практические навыки реализации классов на java.

#### Основное содержание работы.

Написать программу, в которой создаются и разрушаются объекты определенного пользователем класса. Выполнить исследование вызовов конструкторов.

#### Краткие теоретические сведения. Класс.

Класс – фундаментальное понятие java. Класс предоставляет механизм для создания объектов. В классе отражены важнейшие концепции объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм.

С точки зрения синтаксиса, класс – это структурированный тип, образованный на основе уже существующих типов.

Функции – это методы класса, определяющие операции над объектом.

Данные – это поля объекта, образующие его структуру. Значения полей определяет состояние объекта.

#### Пример.

```
class mydate {                                // дата
private int month,day,year;                  // поля: месяц, день, год
void set(int m, int d, int y) {
    month = m;
    day=d;
    year=y;
}                                              // метод – установить дату
// методы – получить дату
public int Month() { return m;}
public int Day() { return d;}
public int Year() { return y;}
void next();                                // метод – установить следующую дату
public void print();                         // метод – вывести дату
}
```

Для описания объекта класса (экземпляра класса) используется конструкция

***имя\_класса имя\_объекта;***

```
mydate today;
mydate my_birthday;
mydate clim[];                               // массив объектов
```

В определяемые объекты входят данные, соответствующие членам – данным класса. Функции – члены класса позволяют обрабатывать данные конкретных объектов класса. Обращаться к данным объекта и вызывать функции для объекта можно с помощью “квалифицированных” имен:

***имя\_объекта.имя\_данного***

*имя\_объекта.имя\_функции*

Например:

```
my_birthday.year=1995;  
my_birthday.month=7;  
my_birthday.day=19;  
today.set(31,12,2014);  
clim[1].print;
```

### **Доступность компонентов класса.**

В рассмотренных ранее примерах классов компоненты классов являются общедоступными. В любом месте программы, где “видно” определение класса, можно получить доступ к компонентам объекта класса. Тем самым не выполняется основной принцип абстракции данных – инкапсуляция (сокрытие) данных внутри объекта. Для изменения видимости компонент в определении класса можно использовать спецификаторы доступа: **public, private**.

Общедоступные (public) компоненты доступны в любой части программы. Они могут использоваться любой функцией как внутри данного класса, так и вне его. Доступ извне осуществляется через имя объекта:

*имя\_объекта.имя\_члена\_класса*

*ссылка\_на\_объект.имя\_члена\_класса*

Собственные (private) компоненты локализованы в классе и не доступны извне. Они могут использоваться функциями – членами данного класса.

Включив механизм инкапсуляции, т.е. скрыв поля от пользователя, мы потеряли возможность применять привычную операцию присваивания для доступа к значениям полей. Кроме того, пришлось создавать методы, заменяющие присваивание.

### **Конструктор.**

Недостатком рассмотренных ранее классов является отсутствие автоматической инициализации создаваемых объектов. Для каждого вновь создаваемого объекта необходимо было вызвать функцию типа либо явным образом присваивать значения данным объекта. Однако для инициализации объектов класса в его определение можно явно включить специальную подпрограмму, называемую **конструктором**. Формат определения конструктора следующий:

*имя\_класса (список\_форм\_параметров)*

```
{  
  операторы_тела_конструктора ;  
}
```

Пример.

```
mydate (int d, int m, int y)  
{  
  day=d; month=m; year=y;  
}
```

Конструктор выделяет память для объекта и инициализирует данные – члены класса.

Программист может создать несколько конструкторов со своими списками параметров. Например, если часто приходится копировать объекты класса, можно создать конструктор копирования, который кроме создания объекта, проинициализирует и значения его полей величинами, полученными от объекта–параметра.

Пример:

```
mydate (mydate Date) {  
    day=Date.day;  
    month=Date.month;  
    year=Date.year;  
}
```

### **Порядок выполнения работы.**

1. Определить пользовательский класс в соответствии с вариантом задания.
2. Все поля класса должны быть закрытыми.
3. Определить в классе следующие конструкторы: без параметров (с установкой значений по умолчанию), с параметрами, копирования.
4. Определить в классе компоненты-функции для просмотра данных, установки полей данных.
5. Написать демонстрационную программу, в которой создаются несколько объектов пользовательского класса и каждый вызов конструктора и функций сопровождается выдачей соответствующего сообщения (какой объект вызвал).
6. **Дополнительное задание (+25% к оценке)** Организовать массив объектов с заданием данных для каждого в цикле с помощью ввода значений с консоли.

### **Содержание отчета.**

1. Титульный лист: название дисциплины; номер и наименование работы; фамилия, имя, отчество студента; дата выполнения.
2. Постановка задачи. Следует дать конкретную постановку, т.е. указать, какой класс должен быть реализован, какие должны быть в нем конструкторы, функции, свойства и т.д.
3. Определение пользовательского класса с комментариями.
4. Реализация конструкторов.
5. Листинг основной программы, в котором должно быть указано, в каком месте и какие функции используются.

### **Варианты заданий.**

#### ***Описания членов - данных пользовательских классов***

##### **1. СТУДЕНТ**

имя – Строка

курс – Целый

пол – char

##### **2. ИЗДЕЛИЕ**

имя – Строка  
шифр – Строка  
количество – Целый

### 3. АДРЕС

имя – Строка  
улица – Строка  
номер дома – Целый

### 4. ЦЕХ

имя – Строка  
начальник – Строка  
количество работающих – Целый

### 5. СТРАНА

имя – Строка  
форма  
правления – Строка  
площадь – Вещественный

### 6. СЛУЖАЩИЙ

имя – Строка  
возраст – Целый  
рабочий стаж – Целый

### 7. БИБЛИОТЕКА

имя – Строка  
автор – Строка  
стоимость – Вещественный

### 8. ТОВАР

имя – Строка  
количество – Целый  
стоимость – Вещественный

### 9. ПЕРСОНА

имя – Строка  
возраст – Целый  
пол – Целый

### 10. ЖИВОТНОЕ

имя – Строка

класс – Строка  
средний вес – Целый

#### 11. КАДРЫ

имя – Строка  
номер цеха – Целый  
разряд – Целый

#### 12. ЭКЗАМЕН

имя студента – Строка  
дата – Целый  
оценка – Целый

#### 13. КВИТАНЦИЯ

номер – Целый  
дата – Целый  
сумма – Вещественный

#### 14. АВТОМОБИЛЬ

марка – Строка  
мощность – Целый  
стоимость – Вещественный

#### 15. КОРАБЛЬ

имя – Строка  
водоизмещение – Целый  
тип – Строка