# Лабораторная работа № 3 студента группы ИТз-51 Черепченко Юрия Сергеевича Выполнение: \_\_\_\_\_\_\_\_\_\_\_ Защита: \_\_\_\_\_\_\_\_\_\_\_

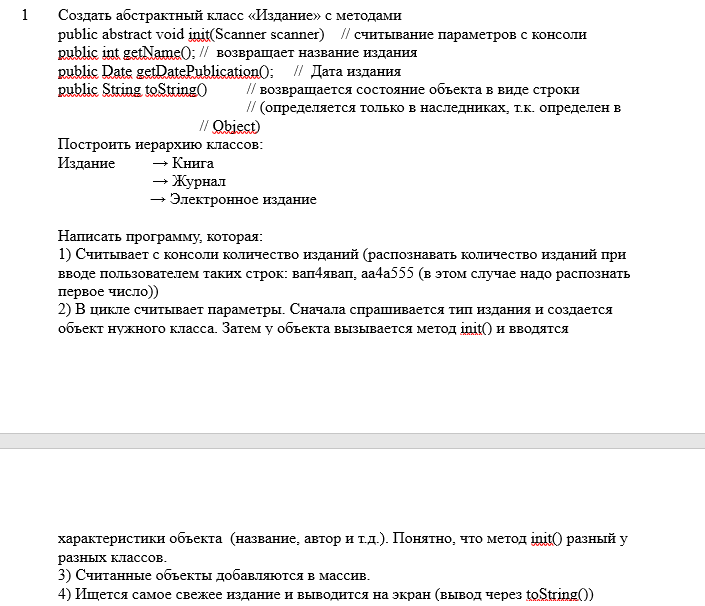
**Основные понятия ООП**

**Цель работы:** изучение основ объектно-ориентированного программирования в Java.

**Вариант-1**

# Задание к лабораторной работе

# Задание



**Текст программы**

**ConsoleMain**

package ru.bstu.itz.cherepchenko.lab1;  
  
  
import java.lang.reflect.Array;  
import java.time.LocalDate;  
import java.util.\*;  
  
public class ConsoleMain {  
  
 List<Publication> publications;  
  
 public Publication getInstance(int type) throws Exception {  
 switch (type) {  
 case 1:  
 return new Book();  
 case 2:  
 return new Magazine();  
 case 3:  
 return new NetworkPublication();  
 default:  
 throw new Exception("Неверный тип класса");  
 }  
 }  
  
 private void start() throws Exception {  
 publications = new ArrayList<Publication>();  
 while (true) {  
 Scanner scanner = new Scanner(System.*in*);  
 int classType = 1;  
 while (classType > 0) {  
 System.*out*.print(  
 "Введите тип издания\n" +  
 "1 - Книга\n" +  
 "2 - Журнал\n" +  
 "3 - Интернет публикация\n" +  
 "0 - Запись\n"  
 );  
 classType = scanner.nextInt();  
 scanner.nextLine();  
 if (classType == 0)  
 break;  
 Publication instance = getInstance(classType);  
 instance.init(scanner);  
 publications.add(instance);  
 }  
  
 Publication minDatePublication = publications.get(0);  
 for (Publication publication : publications)  
 if (publication.getDatePublication().isAfter(minDatePublication.getDatePublication()))  
 minDatePublication = publication;  
 System.*out*.printf(minDatePublication.toString() + ". Самое свежее издание. \n");  
 for (Publication publication : publications)  
 System.*out*.printf(publication.toString() + "\n");  
 }  
 }  
  
 public static void main(String[] args) throws Exception {  
 new ConsoleMain().start();  
 }  
}

**Book**

package ru.bstu.itz.cherepchenko.lab1;  
  
import java.util.Scanner;  
  
public class Book extends Publication {  
 String author;  
 String publishingHouse;  
  
 @Override  
 public void init(Scanner scanner) {  
 name = getString(scanner, "Название");  
 dataPublication = getDate(scanner, "Дату публикации");  
 publishingHouse = getString(scanner, "Издательство");  
 author = getString(scanner, "Автора");  
 }  
  
 @Override  
 public String toString() {  
 return  
 author + " " + name + "( " + dataPublication + " " + publishingHouse + " ) Книга";  
 }  
}

**Magazine**

package ru.bstu.itz.cherepchenko.lab1;  
  
import java.util.Scanner;  
  
public class Magazine extends Publication {  
 String mainEditor;  
 float rating;  
  
 @Override  
 public void init(Scanner scanner) {  
 name = getString(scanner, "Название");  
 dataPublication = getDate(scanner, "Дату публикации");  
 rating = getFloat(scanner, "Рэйтинг журнала");  
 mainEditor = getString(scanner, "Главного редактора");  
 }  
  
 @Override  
 public String toString() {  
 return  
 name + "( " + dataPublication + " " + mainEditor + " ) Журнал";  
 }  
}

**NetworkPublication**

package ru.bstu.itz.cherepchenko.lab1;  
  
import java.util.Scanner;  
  
public class NetworkPublication extends Publication {  
 String url;  
 float averageVisitCount;  
  
 @Override  
 public void init(Scanner scanner) {  
 name = getString(scanner, "Название");  
 dataPublication = getDate(scanner, "Дату публикации");  
 url = getString(scanner, "URL");  
 averageVisitCount = getFloat(scanner, "Среднее количество посещений");  
 }  
  
 @Override  
 public String toString() {  
 return  
 name + "( " + dataPublication + " " + url + " ) Электронное издание";  
 }  
}

**Publication**

package ru.bstu.itz.cherepchenko.lab1;  
  
import java.time.LocalDate;  
import java.time.format.DateTimeFormatter;  
import java.util.Date;  
import java.util.Scanner;  
  
abstract public class Publication {  
 String name;  
 LocalDate dataPublication;  
  
 protected String getString(Scanner scanner, String field){  
 System.*out*.printf("Введите " + field + ": ");  
 return scanner.nextLine();  
 }  
  
 protected float getFloat(Scanner scanner, String field){  
 System.*out*.printf("Введите " + field + ": ");  
 return Float.*parseFloat*(scanner.nextLine());  
 }  
  
 protected LocalDate getDate(Scanner scanner, String field){  
 System.*out*.printf("Введите " + field + ": ");  
 String strDate = scanner.nextLine();  
  
 DateTimeFormatter inFormatter = DateTimeFormatter.*ofPattern*("yyyy-MM-dd");  
 return LocalDate.*parse*(strDate, inFormatter);  
 }  
  
 public abstract void init(Scanner scanner);  
  
 public String getName() {  
 return name;  
 }  
  
 public LocalDate getDatePublication() {  
 return dataPublication;  
 }  
}

**Вывод:**  изучил основы объектно-ориентированного программирования в Java.