Образовательное частное учреждение
Дополнительного профессионального образования «Центр компьютерного обучения «Специалист» Учебно-научного центра при МГТУ им. Н.Э. Баумана»
(ОЧУ «Специалист»)

123317 Москва, Пресненская набережная, д 8, стр. 1, этаж 48, помещение 484с, комната 5

ИНН 7701257303, ОГРН 1037739408189

Утверждаю:
Директор ОЧУ «Специалист»

_____ /Т.С.Григорьева/
«14» февраля 2018 года

Дополнительная профессиональная программа повышения квалификации

«Тестирование ПО. Уровень 1. Тестировщик программного обеспечения»

город Москва

Программа разработана в соответствии с приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. N 499 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам".

Повышение квалификации слушателей, осуществляемое в соответствии с программой, проводится с использованием модульного принципа построения учебного плана с применением различных образовательных технологий, в том числе дистанционных образовательных технологий и электронного обучения в соответствии с законодательством об образовании.

Дополнительная профессиональная программа повышения квалификации, разработана образовательной организацией в соответствии с законодательством Российской Федерации, включает все модули, указанные в учебном плане.

Содержание оценочных и методических материалов определяется образовательной организацией самостоятельно с учетом положений законодательства об образовании Российской Федерации.

Структура дополнительной профессиональной программы соответствует требованиям Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденного приказом Минобрнауки России от 1 июля 2013 г. N 499.

Объем дополнительной профессиональной программы вне зависимости от применяемых образовательных технологий, должен быть не менее 16 академических часов. Сроки ее освоения определяются образовательной организацией самостоятельно.

Формы обучения слушателей (очная, очно-заочная, заочная) определяются образовательной организацией самостоятельно.

К освоению дополнительных профессиональных программ допускаются:

- лица, имеющие среднее профессиональное и (или) высшее образование;
- лица, получающие среднее профессиональное и (или) высшее образование.

Для определения структуры дополнительной профессиональной программы и трудоемкости ее освоения может применяться система зачетных единиц. Количество зачетных единиц по дополнительной профессиональной программе устанавливается организацией.

Образовательная деятельность слушателей предусматривает следующие виды учебных занятий и учебных работ: лекции, практические и семинарские занятия, лабораторные работы, круглые столы, мастер-классы, мастерские, деловые игры, ролевые игры, тренинги, семинары по обмену опытом, выездные занятия, консультации, выполнение аттестационной, дипломной, проектной работы и другие виды учебных занятий и учебных работ, определенные учебным планом.

1. Цель программы:

В результате прохождения обучения слушатель научится тестированию программного продукта, ознакомится с целями тестирования и местом в процессе разработки программного обеспечения, основным методам тестирования программного продукта, работать с документами, применяемыми на этапах планирования и выполнения тестирования, разрабатывать планы тестирования, описывать обнаруженные дефекты (баг-репорты)

Совершенствуемые компетенции

№	Компетенция	Направление подготовки		
		ФГОС	ВО	ПО

		НАПРАВЛЕНИЮ ПОДГОТОВКИ 09.03.02 «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ» (УРОВЕНЬ БАКАЛАВРИАТА)
		Код компетенции
1	способностью проводить моделирование процессов и систем	ПК-5
2	способностью поддерживать работоспособность информационных систем и технологий в заданных функциональных характеристиках и соответствии критериям качества	ПК-30
3	способностью обеспечивать безопасность и целостность данных информационных систем и технологий	ПК-31

Совершенствуемые компетенции в соответствии с трудовыми фикциями профессионального стандарта «ПРОГРАММИСТ», утвержденного приказом Минтруда и социальной защиты РФ от 18 ноября 2013 г. N 679н

№	Компетенция	Направление подготовки
		Трудовые функции (код)
1	Разработка и отладка программного кода (Формализация и алгоритмизация поставленных задач, Написание программного кода с использованием языков программирования, определения и манипулирования данными, Оформление программного кода в соответствии с установленными требованиями)	A/01.3; A/02.3; A/03.3

Планируемый результат обучения:

После окончания обучения Слушатель будет знать:

- Владеть терминологией
- Понимать процесс тестирования программного обеспечения и жизненный цикл программного продукта

После окончания обучения Слушатель будет уметь:

- Понимать процесс тестирования программного обеспечения и жизненный цикл программного продукта
- Разрабатывать тестовые планы (Test Plan) и тестовые примеры (Test Case)
- Выполнять тестирование в соответствии с заранее подготовленным тестовым планом
- Обнаруживать ошибки при выполнении тестирования и документировать их
- Оценивать и тестировать программный продукт с точки зрения функциональности

Учебный план:

Категория слушателей: курс предназначен для:

- Руководители проектов и отделов;
- Руководители линейных подразделений;
- Помощники руководителя проектов
- Программисты
- Разработчики и аналитики компьютерных систем

Требования к предварительной подготовке:

Успешное окончание курса «Основы программирования и баз данных», «Microsoft Excel 2016/2013. Уровень 1. Работа с Excel 2016/2013»

Срок обучения: 40 академических часов, 20 самостоятельно

Форма обучения: очная, очно-заочная, заочная. По желанию слушателя форма обучения может быть изменена и/или дополнена.

Режим занятий: дневной, вечерний, группы выходного дня.

№ п/п	Наименование модулей по программе	Общая трудоемкость (акад. часов)	Всего ауд.ч	В том числе		СРС,ч	ПА*
				Лекций	Практ занятий		
1	Модуль 1. **Введение в тестирование программного обеспечения**	6	4	2	2	2	Уст. пр.
2	Модуль 2. **Методы и виды тестирования. Анализ требований к ПО**	6	4	2	2	2	Уст. пр
3	Модуль 3. **Тестовая документация. Тест-план, тест-дизайн**	6	4	1	3	2	Уст. пр
4	Модуль 4. **Тестовая документация. Test Case. Отчет о прохождении тестов**	6	4	2	2	2	Уст. пр
5	Модуль 5. **Техники тестирования**	6	4	2	2	2	Уст. пр
6	Модуль 6. **Уровни тестирования. Критерии покрытия кода программы тестами**	6	4	2	2	2	Уст. пр
7	Модуль 7. **Уровни тестирования. Критерии покрытия кода программы тестами**	6	4	2	2	2	Уст. пр
8	Модуль 8. **Тестирование пользовательского интерфейса (GUI). Тестирование web-приложений**	6	4	2	2	2	Уст. пр
9	Модуль 9. **Регрессионное тестирование**	6	4	2	2	2	Уст. пр
10	Модуль 10. **Практическая работа по тестированию ПО**	6	4	2	2	2	Уст. пр
	Итог:	72	40	19	21	20	Уст. пр
	Итоговая аттестация	тестирование					

Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут. Форма Промежуточной аттестации – см. в ЛНА «Положение о проведении промежуточной аттестации слушателей и осуществлении текущего контроля их успеваемости» п.3.3

2. Календарный учебный график

Календарный учебный график формируется при осуществлении обучения в течение всего календарного года. По мере набора групп слушателей по программе составляется календарный график, учитывающий объемы лекций, практики, самоподготовки, выезды на объекты.

Неделя обучения	1	2	3	4	5	6	7	Итого часов
	пн	вт	ср	чт	пт	сб	вс	
1 неделя	8	8	8	8	8	-	-	40
СРС	4	4	4	4	4			20
Итого:								60
Примечание: ИА – Итоговая аттестация (лабораторная работа, контрольные вопросы)								

Рабочие программы учебных предметов

Модуль 1. Введение в тестирование программного обеспечения

- Зачем нужно тестировать программы?
- Понятие качества ПО. Стандарты качества ПО.
- Атрибуты и характеристики качества ПО.
- Основные определения тестирования.
- Цели и задачи процесса тестирования.
- Полный цикл тестирования. Фазы тестирования.

Модуль 2. Методы и виды тестирования. Анализ требований к ПО

- Методы и виды тестирования. Общий обзор.
- Критерии покрытия тестирования.
- Требования к ПО.
- Анализ требований с точки зрения пригодности к тестированию.
- Учебный проект: тестирование требований к учебной программе.

Модуль 3. Тестовая документация. Тест-план, тест-дизайн

- Документы, создаваемые в процессе тестирования.
- Тест план
- Связь тестовых планов с другими типами документов.
- Тест – дизайн.
- Возможные формы подготовки тест-дизайна.
- Учебный проект: составление плана тестирования учебной программы.

Модуль 4. Тестовая документация. Test Case. Отчет о прохождении тестов

- Определение Test Case.
- Правила написания, степень детализации, независимость.
- Правила описания дефектов, понятие важности, приоритета.
- Ведение системы отслеживания дефектов.
- Составление отчетов по результатам тестирования.
- Учебный проект: создание test cases для учебной программы.

Модуль 5. Техники тестирования

- Покрытие входных данных. Допустимые и недопустимые данные.
- Эквивалентное разбиение.
- Анализ граничных значений.
- Попарное комбинирование.
- Предположение ошибок.
- Учебный проект: составление набора входных данных для тестирования учебной программы

Модуль 6. Уровни тестирования. Критерии покрытия кода программы тестами

- Модульное тестирование. Драйверы. Заглушки.
- Интеграционное тестирование. Способы интеграционного тестирования.
- Системное тестирование.
- Понятие покрытия кода тестами. Критерии покрытия.
- Метрика покрытия.
- Анализ покрытия.

Модуль 7. Виды тестирования: функциональное и нефункциональное тестирование

- Функциональные виды тестирования.
- Тестирование безопасности, тестирование взаимодействия.
- Нефункциональные виды тестирования.
- Тестирование производительности.
- Нагрузочное тестирование.
- Учебный проект: инсталляционное тестирование учебной программы.

Модуль 8. Тестирование пользовательского интерфейса (GUI). Тестирование web-приложений

- Задачи и цели тестирования пользовательского интерфейса.
- Функциональное тестирование пользовательского интерфейса.
- Тестирование удобства пользовательского интерфейса.
- Тестирование web-приложений
- Учебный проект: функциональное тестирование GUI

Модуль 9. Регрессионное тестирование

- Регрессионное тестирование. Подходы к составлению набора test cases.
- Жизненный цикл ПО. Каскадный, спиральный жизненные циклы.
- Методологии разработки ПО. MSF, RUP, Экстремальное программирование.
- Команда тестирования. Роли.

Модуль 10. Практическая работа по тестированию ПО

- Изучение требований к ПО.
- Написание тестовой документации (плана тестирования и тестовых сценариев)
- Выполнение тестирования
- Написание отчетов о найденных дефектах.
- Подведение итогов и обсуждение результатов слушателей

Организационно-педагогические условия

Соблюдение требований к кадровым условиям реализации дополнительной профессиональной программы:

а) преподавательский состав образовательной организации, обеспечивающий образовательный процесс, обладает высшим образованием и стажем преподавания по изучаемой тематике не менее 1 года и (или) практической работы в областях знаний, предусмотренных модулями программы, не менее 3 (трех) лет;

б) образовательной организацией наряду с традиционными лекционно-семинарскими занятиями применяются современные эффективные методики преподавания с применением интерактивных форм обучения, аудиовизуальных средств, информационно-телекоммуникационных ресурсов и наглядных учебных пособий.

Соблюдение требований к материально-техническому и учебно-методическому обеспечению дополнительной профессиональной программы:

а) образовательная организация располагает необходимой материально-технической базой, включая современные аудитории, библиотеку, аудиовизуальные средства обучения, мультимедийную аппаратуру, оргтехнику, копировальные аппараты. Материальная база соответствует санитарным и техническим нормам и правилам и обеспечивает проведение всех видов практической и дисциплинарной подготовки слушателей, предусмотренных учебным планом реализуемой дополнительной профессиональной программы.

б) в случае применения электронного обучения, дистанционных образовательных технологий каждый обучающийся в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом к электронной информационно-образовательной среде, содержащей все электронные образовательные ресурсы, перечисленные в модулях дополнительной профессиональной программы.

3. Формы аттестации и оценочные материалы

Образовательная организация несет ответственность за качество подготовки слушателей и реализацию дополнительной профессиональной программы в полном объеме в соответствии с учебным планом.

Оценка качества освоения дополнительной профессиональной программы слушателей включает текущий контроль успеваемости и итоговую аттестацию.

Конкретные формы и процедуры текущего контроля успеваемости, промежуточной аттестации и итоговой аттестации слушателей устанавливаются образовательной организацией самостоятельно.

Слушателям, успешно освоившим дополнительную профессиональную программу и прошедшим итоговую аттестацию, выдается удостоверение о повышении квалификации.

Слушателям, не прошедшим итоговой аттестации или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть дополнительной профессиональной программы и (или) отчисленным из образовательной организации, выдается справка об обучении или о периоде обучения по образцу, самостоятельно устанавливаемому образовательной организацией.

Результаты итоговой аттестации слушателей ДПП в соответствии с формой итоговой аттестации, установленной учебным планом, выставляются по двух бальной шкале («зачтено\незачтено»).

Итоговая аттестация проводится по форме выполнения лабораторной работы и аттестации по контрольным вопросам в соответствии с учебным планом. Результаты итоговой аттестации заносятся в соответствующие документы.

Промежуточная аттестация

В. Что такое динамическое тестирование?

О. Это тестирование за счет выполнения кода или программы с различными входными значениями и подтверждением результатов.

В. Что такое GUI-тестирование (GUI Testing)?

О. Тестирование GUI (графического интерфейса пользователя): интерфейс программного обеспечения проверяется на предмет соответствия требованиям.

В. Что такое формальное тестирование?

О. Верификация программного обеспечения, согласно тест-плану, тестовым процедурам и соответствующей документации, с учетом пожеланий клиента.

В. Что такое тестирование на основе рисков?

О. Определяются наиболее важные части системы, затем устанавливается порядок их тестирования, затем следует, собственно, тестирование.

В. Что такое раннее тестирование?

О. Тестирование по возможности проводится как можно раньше, чтобы выявить дефекты на ранних этапах SDLC. Это позволяет быстрее обнаружить и устранить дефекты, экономит расходы.

В. Что такое исчерпывающее тестирование?

О. Тестирование функциональности, с использованием неверных и верных данных ввода и входных условий.

В. Что такое скопление дефектов?

О. Даже небольшой модуль или функциональность могут содержать в себе ряд дефектов, поэтому необходимо больше уделять внимания тестированию функциональности.

В. Что такое «парадокс пестицида»?

О. Если с помощью имеющихся тестовых сценариев не получается обнаружить дефекты, возможно, стоит дополнить/пересмотреть тест-кейсы, чтобы можно было находить больше дефектов.

В. Что такое статическое тестирование?

О. Верификация кода вручную без программы. В этом процессе проблемы находятся в коде, во время его проверки и сравнения с требованиями.

В. Что такое позитивное тестирование?

О. Тестирование, которое проводится в приложении с целью определить, насколько система функциональна. Такой подход больше известен как «тест на прохождение».

Итоговая аттестация

Лабораторная работа. Структура программы, основные типы данных, ввод/вывод

Цель работы – изучить структуру программы, научиться использовать переменные различных типов, освоить функции форматного ввода и вывода, арифметические операции и операции присваивания.

Постановка задачи

1. Набрать текст программы, представленный ниже. Проанализировать значения переменных после каждой операции присваивания. Проверить порядок выполнения операций в каждом выражении, содержащем несколько операций присваивания, разделив каждый оператор-выражение на несколько операторов, выполняемых последовательно. В функциях ввода и вывода изменить спецификаторы формата, проанализировать полученные результаты.

```
#include <stdio.h>

#include <stdlib.h>

int main (void)

{

int a, b = 5, c;

double x, y = -.5, z;
```

```
printf("a=");
scanf("%d", &a);
x = c = a;
printf("x = c = a : a=%d c=%d x=%f\n", a, c, x);
a += b;
printf("a += b : a=%d\n", a);
x *= b+a;
printf("x *= b+a : x=%lf\n", x);
b += a--;
printf("b += a-- : a=%d b=%d\n", a, b);
x -= ++c;
printf("x -= ++c : c=%d x=%lf\n", c, x);
c = a/b;
printf("c = a/b : c=%4d\n",c);
c = a%b;
printf("c = a%%b : c=%d\n",c);
y += (a+1)/a++;
printf("y += (a+1)/a++ : a=%d y=%.3lf\ty=%.0lf\n", a, y, y);
b = 3*(y-=.6)+2*b+1;
printf("b = 3*(y-=.6)+2*b+1 : b=%d y=%.1lf\n", b, y);
z = a/2;
printf("z = a/2 : z = a/2 : z=%lf\n", z);
z = (double)a/2;
printf("z = (double)a/2 : z=%lf\n", z);
y = (x = 5.7)/2;
printf("y = (x = 5.7)/2 : x=%lf y=%lf\n", x, y);
y = (int)x/2;
printf("y = (int)x/2 : y=%f\n", y);
```

```
z = (b-3)/2 - x/5 +(c/=2) + 1/4*z - y++ + ++b/3.;

printf("z = (b-3)/2 - x/5 +(c/=2) + 1/4*z - y++ + ++b/3. :\n\

a=%d b=%d c=%d x=%lf y=%lf z=%lf\n", a,b,c,x,y,z);

system("pause");

return 0;

}
```

2. Написать программу для вычисления значений следующих выражений: a=5,c=5

```
a=a+b-2

c=c+1, d=c-a+d

a=a*c, c=c-1

a=a/10, c=c/2, b=b-1, d=d*(c+b+a)
```

Выражения, записанные в одной строке, записывать одним оператором-выражением. Переменные cидобъявить как целые, переменныеаиb– как вещественные. Значения переменныхbидвводить с клавиатуры. После вычисления каждого выражения выводить на экран значения всех переменных.

3. Набрать текст программы, представленный ниже. Проанализировать выдаваемые программой результаты. Объяснить, почему они именно такие.

```
#include <stdlib.h>

#include <stdio.h>

#include <limits.h>

#include <float.h>

main()

{

char c;

unsigned char uc;

int i;

unsigned u;

short s;

long l;

float f;
```

```
double d;
printf("sizeof(c)=%d\tsizeof(uc)=%d\nsizeof(i)=%d\tsizeof(u)=%d\t\
sizeof(s)=%d\tsizeof(l)=%d\nsizeof(f)=%d\tsizeof(d)=%d\n\n",
sizeof(c),sizeof(uc),sizeof(i),sizeof(u),sizeof(s),
sizeof(l),sizeof(f),sizeof(d));
uc=c=CHAR_MAX;
printf("CHAR_MAX : c=%d uc=%d\n", c, uc);
c++; uc++;
printf("CHAR_MAX+1 : c=%d uc=%d\n", c, uc);
uc=c=CHAR_MIN;
printf("CHAR_MIN : c=%d uc=%d\n", c, uc);
c=uc=UCHAR_MAX;
printf("UCHAR_MAX : c=%d uc=%d\n", c, uc);
c++; uc++;
printf("UCHAR_MAX+1 : c=%d uc=%d\n", c, uc);
uc=c=-5;
printf("-5 : c=%d uc=%d\n", c, uc);
c=-5; uc=5;
printf("char and unsigned char -5>5 : %d\n\n", c>uc);
c=s=SHRT_MAX;
uc=s;
printf("SHRT_MAX : c=%d uc=%d s=%d\n", c, uc, s);
s++;
printf("SHRT_MAX+1 : s=%d\n", s);
c=s; uc=s;
printf("%d : c=%d uc=%d\n", SHRT_MIN, c, uc);
s=0; c=s; uc=s;
printf("0 : c=%d uc=%d s=%d\n", c, uc, s);
```

```
i=INT_MAX;
l=i; u=i;
printf("INT_MAX : i=%d u=%u l=%ld\n", i, u, l);
i++; l++; u++;
printf("INT_MAX+1 : i=%d u=%u l=%ld\n", i, u, l);
i=INT_MIN;
l=i; u=i;
printf("INT_MIN : i=%d u=%u l=%ld\n", i, u, l);
u=UINT_MAX;
i=u; l=u;
printf("UINT_MAX : i=%d u=%u l=%ld\n", i, u, l);
u=i=-5;
printf("-5 : i=%d u=%u\n", i, u);
i=-5; u=5;
printf("int and unsigned int -5>5 : %d\n", i>u);
c=-5; u=5;
printf("char and unsigned int -5>5 : %d\n\n", c>u);
i=5.1;
printf("i=5.1 : i=%d\n", i);
i=5.9;
printf("i=5.9 : i=%d\n", i);
d=f=FLT_MAX;
printf("FLT_MAX : f=%g d=%g\n", f, d);
d=f=FLT_MIN;
printf("FLT_MIN : f=%g d=%g\n", f, d);
d=f=FLT_EPSILON;
printf("FLT_EPSILON : f=%g d=%g\n", f, d);
f=1e10;
```

```
printf("1e10 : f=%f\n", f);
f=1e11;
printf("1e11 : f=%f\n", f);
f=1234567890;
printf("1234567890 : f=%f\n", f);
d=DBL_MAX;
printf("DBL_MAX : d=%g\n", d);
d=DBL_MIN;
printf("DBL_MIN : d=%g\n", d);
d=DBL_EPSILON;
printf("DBL_EPSILON : d=%g\n", d);
d=1e15+1;
printf("1e15+1 : d=%lf\n", d);
d=1e16+1;
printf("1e16+1 : d=%lf\n", d);
f=10000*100000;
f+=1;
f-=4*250000000;
printf("1 : f=%f\n", f);
f=10000*100000+1-4*250000000;
printf("1 : f=%f\n", f);
d=10000*100000+1-4*250000000;
printf("1 : d=%lf\n", d);
d=1e20*1e20+1000-1e22*1e18;
printf("1000 : d=%lf\n", d);
system("pause");
return 0;
}
```

Контрольные вопросы

1. Какова структура программы на языке Си?
2. Зачем нужна директива #include?
3. Что такое main()?
4. Перечислите скалярные типы данных языка Си.
5. Что определяет тип данного?
6. Что такое void?
7. Что такое явное и неявное приведение типов? Как и когда оно используется?
8. Что такое константа? Найдите константы в набранных вами программах.
9. Что такое переменная?
10. Как проинициализировать переменную?
11. Чем отличается оператор от операции?
12. Чем отличаются унарные операции от бинарных?
13. Какие операции относятся к арифметическим? Каков приоритет каждой из них?
14. Каков порядок выполнения операций в случае их одинакового приоритета?
15. Как выполняется операция деления в случае целочисленных операндов и в случае, когда хотя бы один из операндов вещественный?
16. Что такое выражение?
17. Какое значение вычисляет операция присваивания?
18. В каком порядке выполняются присваивания в случае, если в выражении их несколько?
19. Как и зачем используются дополнительные операции присваивания?
20. Чем отличается префиксная форма операции инкремента или декремента от постфиксной?
21. Какие функции используются для ввода информации? Назовите их отличительные особенности.
22. Какие функции используются для вывода информации? Назовите их отличительные особенности.
23. Почему функции scanf() иprintf() называются функциями форматного ввода и вывода? Как они работают?
24. Чем отличается управляющая строка функции scanf() от управляющей строки функцииprintf()?
25. Что такое спецификатор формата? Зачем он нужен?
26. Какие параметры указываются функции scanf() после управляющей строки? Сколько их должно быть?
27. Каковы последствия несоответствия типа считываемой функцией scanf() переменной спецификатору типа?
28. Какие параметры указываются функции printf() после управляющей строки? Сколько их должно быть?
29. Каковы последствия несоответствия типа выводимого функцией printf() значения спецификатору типа?
30. Что такое управляющие символы? Зачем они нужны? Приведите примеры.